

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

NUVEI TECHNOLOGIES, INC.,

Petitioners,

v.

AUTOSCRIBE CORPORATION,

Patent Owners.

Case IPR2025-00089
U.S. Patent No. 11,620,6121

DECLARATION OF KEVIN C. ALMEROTH, Ph.D.

TABLE OF CONTENTS

I. INTRODUCTION	1
II. PROFESSIONAL BACKGROUND.....	1
III. COMPENSATION	13
IV. LEGAL CONSIDERATION.....	13
A. Obviousness.....	14
B. Claim Interpretation	16
V. OVERVIEW OF TASK & BASIS FOR OPINIONS	17
VI. PERSON OF ORDINARY SKILL IN THE ART	18
A. A POSITA’s Ordinary Level of Skill.....	18
VII. A POSITA’S UNDERSTANDING OF THE FIELD OF TECHNOLOGY	
19	
A. HTTP Communications.....	19
1. HTTP Communications.....	19
2. Conveying State/Context Information In Stateless HTTP	20
3. HTML and iFrames	21
4. HTTPS / Secure Socket Layer (“SSL”) Connections	22
5. Application Programming Interface (API).....	22

B.	Online Shopping and Payments	24
1.	Online Shopping and Payment Processing	24
2.	Tokenization	26
VIII.	OVERVIEW OF THE '621 PATENT	27
IX.	GROUND 1: STRINGFELLOW AND KLOSTER	29
A.	U.S. Patent Application Pub. No. 2010/0175626 A1 (“Stringfellow”)	29
1.	Online Merchant System → Trusted Intermediary: Originating Payment Authorisation Request Message.	31
2.	Trusted Intermediary → Online Merchant System: Generation and Retrieval Of Sign-In Page URL.....	32
3.	Online Merchant System → Client: Provide Sign-In Page URL Via iFrame; Client ↔ Trusted Intermediary: Request URL, Establish SSL Session, Serve Sign-In Page.	32
4.	Client ↔ Trusted Intermediary: Account Registration, Including Collecting and Storing Sensitive Financial Information.	34
5.	Content of “Sign-In URL.”	36
B.	U.S. Patent Application Pub. No. 2010/0094755 A1 (“Kloster”)	38
C.	Motivation to Combine Stringfellow and Kloster.....	40

D.	Modification of Stringfellow in view of Kloster	43
E.	Element-by-Element Analysis.....	46
1.	Claim 1	46
2.	Claim 2	65
3.	Claim 3	66
4.	Claim 4	66
5.	Claim 5	66
6.	Claim 6	67
7.	Claim 7	68
8.	Claim 8	68
9.	Claim 9	74
10.	Claim 10	75
11.	Claim 11	75
12.	Claim 12	76
13.	Claim 13	76
14.	Claim 14	76
15.	Claim 15	76

16. Claim 16	81
17. Claim 17	81
18. Claim 18	81
19. Claim 19	82
20. Claim 20	82
21. Claim 21	82
22. Claim 22	82
23. Claim 23	83
24. Claim 24	86
25. Claim 25	86
26. Claim 26	90
27. Claim 27	91
X. GROUND 2: STRINGFELLOW, KLOSTER, AND CARLSON.....	91
A. Carlson.....	91
B. Motivation to Combine	93
C. Modification of Stringfellow in view of Kloster and Carlson	94
D. Element-By-Element Analysis.....	94

XI. CONCLUSION.....	95
---------------------	----

I. INTRODUCTION

1. My name is Dr. Kevin C. Almeroth, Ph.D. I have been retained by the Petitioners (“Nuvei”) as an independent expert consultant in this inter partes review (“IPR”) proceeding before the United States Patent and Trademark Office (“PTO”) regarding US Patent No. 11,620,621 (“the ’621 Patent”).

II. PROFESSIONAL BACKGROUND

2. All of my opinions stated in this Declaration are based on my own personal knowledge and professional judgment. In forming my opinions, I have relied on my knowledge and experience in designing, developing, researching, and teaching the technology referenced in this Declaration.

3. I am over 18 years of age and, if I am called upon to do so, I would be competent to testify as to the matters set forth herein. I understand that a copy of my current curriculum vitae, which details my education and professional and academic experience, is being submitted as EX2002. The following provides a brief overview of some of my experience that is relevant to the matters set forth in this Declaration.

4. I am currently a Professor Emeritus in the Department of Computer Science at the University of California, Santa Barbara (UCSB). While active at UCSB, I held faculty appointments and was a founding member of the Computer Engineering (CE) Program, Media Arts and Technology (MAT) Program, and the

Technology Management Program (TMP). I also served as the Associate Director of the Center for Information Technology and Society (CITS) from 1999 to 2012. I have been a faculty member at UCSB since July 1997.

5. I hold three degrees from the Georgia Institute of Technology: (1) a Bachelor of Science degree in Information and Computer Science (with minors in Economics, Technical Communication, and American Literature) earned in June 1992; (2) a Master of Science degree in Computer Science (with specialization in Networking and Systems) earned in June 1994; and (3) a Doctor of Philosophy (Ph.D.) degree in Computer Science (Dissertation Title: Networking and System Support for the Efficient, Scalable Delivery of Services in Interactive Multimedia System, minor in Telecommunications Public Policy) earned in June 1997. During my education, I have taken a wide variety of courses as demonstrated by my minors. My undergraduate degree also included a number of courses more typical of a degree in electrical engineering including digital logic, signal processing, and telecommunications theory.

6. One of the major concentrations of my research over the past 30+ years has been the delivery of multimedia content and data between computing devices, including various network architectures. In my research, I have studied large-scale content delivery systems, and the use of servers located in a variety of geographic locations to provide scalable delivery to hundreds or thousands of users

simultaneously. I have also studied smaller-scale content delivery systems in which content is exchanged between individual computers and portable devices. My work has emphasized the exchange of content more efficiently across computer networks, including the scalable delivery of content to many users, mobile computing, satellite networking, delivering content to mobile devices, and network support for data delivery in wireless networks.

7. In 1992, the initial focus of my research was on the provision of interactive functions (e.g., VCR-style functions like pause, rewind, and fast-forward) for near video-on-demand systems in cable systems; in particular, how to aggregate requests for movies at a cable head-end and then how to satisfy a multitude of requests using one audio/video stream broadcast to multiple receivers simultaneously. This research has continually evolved and resulted in the development of techniques to scalably deliver on-demand content, including audio, video, web documents, and other types of data, through the Internet and over other types of networks, including over cable systems, broadband telephone lines, and satellite links.

8. An important component of my research has been investigating the challenges of communicating multimedia content, including video, between computers and across networks including the Internet. Although the early Internet was used mostly for text-based, non-real time applications, the interest in sharing

multimedia content, such as video, quickly developed. Multimedia-based applications ranged from downloading content to a device to streaming multimedia content to be instantly used. One of the challenges was that multimedia content is typically larger than text-only content, but there are also opportunities to use different delivery techniques since multimedia content is more resilient to errors. I have worked on a variety of research problems and used a number of systems that were developed to deliver multimedia content to users. One content-delivery method I have researched is the one-to-many communication facility called “multicast,” first deployed as the Multicast Backbone, a virtual overlay network supporting one-to-many communication. Multicast is one technique that can be used on the Internet to provide streaming media support for complex applications like video-on-demand, distance learning, distributed collaboration, distributed games, and large-scale wireless communication. The delivery of media through multicast often involves using Internet infrastructure, devices and protocols, including protocols for routing and TCP/IP.

9. Starting in 1997, I worked on a project to integrate the streaming media capabilities of the Internet together with the interactivity of the web. I developed a project called the Interactive Multimedia Jukebox (IMJ). Users would visit a web page and select content to view. The content would then be scheduled on one of a number of channels, including delivery to students in Georgia Tech dorms delivered

via the campus cable plant. The content of each channel was delivered using multicast communication.

10. In the IMJ, the number of channels varied depending on the capabilities of the server including the available bandwidth of its connection to the Internet. If one of the channels was idle, the requesting user would be able to watch their selection immediately. If all channels were streaming previously selected content, the user's selection would be queued on the channel with the shortest wait time. In the meantime, the user would see what content was currently playing on other channels, and because of the use of multicast, would be able to join one of the existing channels and watch the content at the point it was currently being transmitted.

11. The IMJ service combined the interactivity of the web with the streaming capabilities of the Internet to create a jukebox-like service. It supported true Video-on-Demand when capacity allowed but scaled to any number of users based on queuing requested programs. As part of the project, we obtained permission from Turner Broadcasting to transmit cartoons and other short-subject content. We also connected the IMJ into the Georgia Tech campus cable television network so that students in their dorms could use the web to request content and then view that content on one of the campus's public access channels.

12. More recently, I have also studied issues concerning how users choose content, especially when considering the price of that content. My research has examined how dynamic content pricing can be used to control system load. By raising prices when systems start to become overloaded (i.e., when all available resources are fully utilized) and reducing prices when system capacity is readily available, users' capacity to pay as well as their willingness can be used as factors in stabilizing the response time of a system. This capability is particularly useful in systems where content is downloaded or streamed on-demand to users.

13. As a parallel research theme, starting in 1997, I began researching issues related to wireless devices and sensors. In particular, I was interested in showing how to provide greater communication capability to "lightweight devices," i.e., small form-factor, resource-constrained (e.g., CPU, memory, networking, and power) devices. Starting in 1998, I published several papers on my work to develop a flexible, lightweight, battery-aware network protocol stack. The lightweight protocols we envisioned were similar in nature to protocols like Bluetooth, Universal Plug and Play (UPnP) and Digital Living Network Alliance (DLNA).

14. From this initial work, I have made wireless networking—including ad hoc, mesh networks and wireless devices—one of the major themes of my research. My work in wireless network spans the protocol stack from applications through to the encoding and exchange of data at the data link and physical layers.

15. At the application layer, even before the large-scale “app stores” were available, my research looked at building, installing, and using apps for a variety of purposes, from network monitoring to support for traditional computer-based applications (e.g., content retrieval) to new applications enabled by ubiquitous, mobile devices. For example, my research has looked at developing applications for virally exchanging and tracking “coupons” through “opportunistic contact” (i.e., communication with other devices coming into communication range with a user). In many of the courses I have taught there is a project component. Through these projects I have supervised numerous efforts to develop new “apps” for download and use across a variety of mobile platforms.

16. Toward the middle of the protocol stack, my research also looked to build wireless infrastructure support to enable communication among a set of mobile devices unaided by any other kind of network infrastructure. These kinds of networks are useful either in challenged network environments (e.g., when a natural disaster has destroyed existing infrastructure) or when suitable support for network communication never existed. The deployment of such networks (or even the use of traditional network support) are critical to support services like disaster relief, catastrophic event coordination, and emergency services deployment.

17. Yet another theme is monitoring wireless networks, in particular different variants of IEEE 802.11 compliant networks, to (1) understand the

operation of the various protocols used in real-world deployments, (2) use these measurements to characterize use of the networks and identify protocol limitations and weaknesses, and (3) propose and evaluate solutions to these problems. I have successfully used monitoring techniques to study wireless data link layer protocol operation and to improve performance by enhancing the operation of such protocols. For wireless protocols, this research includes functions like network acquisition and channel bonding. Protecting networks, including their operation and content, has been an underlying theme of my research almost since the beginning of my research career. Starting in 2000, I have been involved in several projects that specifically address security, network protection, and firewalls. After significant background work, a team on which I was a member successfully submitted a \$4.3M grant proposal to the Army Research Office (ARO) at the Department of Defense to propose and develop a high-speed intrusion detection system. Key aspects of the system included associating streams of packets and analyzing them for viruses and other malware. Once the grant was awarded, we spent several years developing and meeting the milestones of the project. A number of my students worked on related projects and published papers on topics ranging from intrusion detection to developing advanced techniques to be incorporated into firewalls. I have also used firewalls, including their associated malware detection features, in developing

techniques for the classroom to ensure that students are not distracted by online content.

18. Recent work ties some of the various threads of my past research together. I have investigated content delivery in online social networks and proposed reputation management systems in large-scale social networks and marketplaces. On the content delivery side, I have looked at issues of caching and cache placement, especially when content being shared, and the cache has geographical relevance. We were able to show that effective caching strategies can greatly improve performance and reduce deployment costs. Our work on reputation systems showed that reputations have economic value, and as such, creates a motivation to manipulate reputations. In response, we developed a variety of solutions to protect the integrity of reputations in online social networks. The techniques we developed for content delivery and reputation management were particularly relevant in peer-to-peer communication and recommendations for downloadable “apps.”

19. As an important component of my research program, I have been involved in the development of academic research into available technology in the marketplace. One aspect of this work is my involvement in the Internet Engineering Task Force (IETF). The IETF is a large and open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. I have been

involved in various IETF groups including many content delivery-related working groups like the Audio Video Transport (AVT) group, the MBone Deployment (MBONED) group, Source Specific Multicast (SSM) group, the Inter-Domain Multicast Routing (IDMR) group, the Reliable Multicast Transport (RMT) group, the Protocol Independent Multicast (PIM) group, etc. I have also served as a member of the Multicast Directorate (MADDOGS), which oversaw the standardization of all things related to multicast in the IETF. Finally, I was the Chair of the Internet2 Multicast Working Group for seven years.

20. My involvement in the research community extends to leadership positions for several academic journals and conferences. I am the co-chair of the Steering Committee for the ACM Network and System Support for Digital Audio and Video (NOSSDAV) workshop and on the Steering Committees for the International Conference on Network Protocols (ICNP), ACM Sigcomm Workshop on Challenged Networks (CHANTS), and IEEE Global Internet (GI) Symposium. I have served or am serving on the Editorial Boards of IEEE/ACM Transactions on Networking, IEEE Transactions on Mobile Computing, IEEE Network, ACM Computers in Entertainment, AACE Journal of Interactive Learning Research (JILR), and ACM Computer Communications Review. I have co-chaired a number of conferences and workshops, including the IEEE International Conference on Network Protocols (ICNP), IEEE Conference on Sensor, Mesh and Ad Hoc

Communications and Networks (SECON), International Conference on Communication Systems and Networks (COMSNETS), IFIP/IEEE International Conference on Management of Multimedia Networks and Services (MMNS), the International Workshop On Wireless Network Measurement (WiNMee), ACM Sigcomm Workshop on Challenged Networks (CHANTS), the Network Group Communication (NGC) workshop, and the Global Internet Symposium, and I have served on the program committees for numerous conferences.

21. Furthermore, in the courses I have taught at UCSB, a significant portion of my curriculum covered aspects of the Internet and network communication, including the physical and data link layers of the Open System Interconnect (OSI) protocol stack, and standardized protocols for communicating across a variety of physical media such as cable systems, telephone lines, wireless, and high-speed Local Area Networks (LANs). The courses I have taught also cover most major topics in Internet communication, including data communication, multimedia encoding, and mobile application design. My research and courses have covered a range of physical infrastructures for delivering content over networks, including cable, Integrated Services Digital Network (ISDN), Ethernet, Asynchronous Transfer Mode (ATM), fiber, and Digital Subscriber Line (DSL). For a complete list of courses I have taught, see my curriculum vitae (CV) (EX2002).

22. In addition, I co-founded a technology company called Santa Barbara Labs that was working under a sub-contract from the U.S. Air Force to develop very accurate emulation systems for the military's next generation internetwork. Santa Barbara Labs' focus was in developing an emulation platform to test the performance characteristics of the network architecture in the variety of environments in which it was expected to operate, and, in particular, for network services including IPv6, multicast, Quality of Service (QoS), satellite-based communication, and security. Applications for this emulation program included communication of a variety of multimedia-based services, including video conferencing and video-on-demand.

23. In addition to having co-founded a technology company myself, I have worked for, consulted with, and collaborated with companies for nearly 30 years. These companies range from well-established companies to start-ups and include IBM, Hitachi Telecom, Turner Broadcasting System (TBS), Bell South, Digital Fountain, RealNetworks, Intel Research, Cisco Systems, and Lockheed Martin.

24. Through my graduate education, leadership with CITS, involvement in TMP, role in the development of the Internet2 infrastructure, and consulting with ISPs, I have gained a strong understanding in the role of the Internet in our society and the challenges of deploying large-scale production networking infrastructure. CITS, since its inception, has looked at the role of the Internet in society, including

how the evolution of technology have created communication opportunities and challenges, including, for example through disruptive technologies like P2P. TMP looks to focus on non-purely technical issues, including, for example, state-of-the-art business methods, strategies for successful technology commercialization, new venture creation, and best practices for fostering innovation. Through my industry collaborations and Internet2 work, I have developed significant experience in the challenges of deploying, monitoring, managing, and scaling communication infrastructure to support evolving Internet services like streaming media, conferencing, content exchange, social networking, and e-commerce.

25. I am a Member of the Association of Computing Machinery (ACM) and a Fellow of the Institute of Electrical and Electronics Engineers (IEEE).

III. COMPENSATION

26. My work on this matter is being billed at my customary rate of \$850 an hour. Also, I am being reimbursed for reasonable expenses I incur in relation to my services. I have no pecuniary interest in the outcome of this proceeding. I understand I will be paid regardless of the outcome of any proceeding in which my work is used.

IV. LEGAL CONSIDERATION

27. I am not an attorney. My understanding of the law is based on information provided by counsel for Nuvei.

A. Obviousness

28. I understand that a claimed invention is obvious and, therefore, not patentable if the subject matter claimed would have been obvious to a person of ordinary skill in the art at the time that the invention of the '621 Patent (a "POSITA"), which I have been asked to treat as June 5, 2012.

29. I understand that a claim can be obvious in view of a single prior art reference (e.g., via modification of that prior art reference) or multiple prior art references (e.g., via a combination of two or more prior art references), if such a modification or combination was within the skill of a POSITA. I understand that there must be some articulated reasoning with some rational underpinning to support a conclusion of obviousness. I also understand that to establish a finding of obviousness one must show that one of ordinary skill in the art would have had a motivation to combine the prior art references to produce the claimed invention and a reasonable expectation that the combination would be successful.

30. I further understand that exemplary rationales that may support a conclusion of obviousness include: (1) simply arranging old elements in a way in which each element performs the same function it was known to perform, and the arrangement yields expected results, (2) merely substituting one element for another known element in the field, if the substitution yields no more than a predictable result, (3) combining elements in a way that was "obvious to try" because of a design

need or market pressure, where there was a finite number of identified, predictable solutions, (4) that design incentives or other market forces in a field would have prompted variations in a work that were predictable to a person of ordinary skill in the art, and (5) that there was some teaching, suggestion, or motivation in the prior art that would have led a POSITA to modify or combine prior art references to arrive at the claimed invention. I further understand that certain factors may support or rebut the obviousness of a claim. I understand that such secondary considerations include, among other things, commercial success of the patented invention, skepticism of those having ordinary skill in the art at the time of the invention, unexpected results of the invention, any long-felt but unsolved need in the art that was satisfied by the alleged invention, the failure of others to make the alleged invention, praise of the alleged invention by those having ordinary skill in the art, and copying of the alleged invention by others in the field. I understand that there must be a nexus—that is, a connection between any such secondary considerations and the alleged invention. I also understand that contemporaneous and independent invention by others is a secondary consideration tending to show obviousness. I am not aware of any allegations by the named inventor of the '621 Patent or any assignee of the '621 Patent that any secondary considerations tend to rebut the obviousness of the '621 Patent.

B. Claim Interpretation

31. I understand that a claim term is interpreted according to its ordinary and customary meaning as a POSITA would have understood the term in light of the surrounding claim language, other claims, the specification, and the patent's prosecution history, which are referred to as intrinsic evidence. I also understand that prior art references cited in the patent's prosecution history are considered intrinsic evidence. I further understand that evidence outside the patent and its prosecution history (e.g., dictionaries and technical articles), may inform the context in which a POSITA would have understood the claims of a patent. I understand this ordinary and customary meaning applies absent unique circumstances, such as where a patent clearly expresses an intent to set forth a special meaning for a term.¹

¹ In IPR2019-00062, three constructions were proposed based on (1) “providing a non-sensitive electronic data token representing the sensitive financial account information to the merchant...without providing the non-sensitive electronic data token to the payer” / “receiving the non-sensitive electronic data token from the secure server, wherein...the secure server does not provide the non-sensitive electronic data token to the payer” being construed to refer to a period “*during execution of the token retrieval function...*,” and (2) “generating a uniform resource locator (URL), ... the URL comprising either: a dynamic URL generated by the secure server...; or a static URL and a hypertext transport protocol (HTTP) parameter...” being construed to include “the secure server ‘generating’ the URL by prompting the merchant server to combine a predefined static URL with an ‘HTTP parameter’...[where] ‘generating a uniform resource locator’ does not strictly require that the URL be generated entirely at the secure server.” IPR2019-00062, Paper 4, 11, 15, 17. I do not believe that any claim term requires explicit construction to resolve the issues presented in this Petition.

V. OVERVIEW OF TASK & BASIS FOR OPINIONS

32. I have been asked to review the '621 Patent. I have been asked to provide opinions related to certain issues from the perspective of a person of ordinary skill, having knowledge of the relevant art, as of June 5, 2012, and—except where otherwise noted—the opinions stated in this declaration are from that perspective. The qualifications and abilities of such a person are described in paragraph 36 below.

33. In preparing this declaration, I have considered this patent and its prosecution history as well as the general knowledge of those familiar with the field of technology relating to secure transactions, including e-commerce transaction over the Web, Internet, as well as network communication, data transfer, and content sharing across networks, including the HTTP/HTTPS and TCP/IP protocols, as of the earliest possible priority date of June 5, 2012.

34. My opinions are based on my education, training, and experience as well as items that I reviewed to prepare my opinions, including the documents in the Exhibit List included with the Petition, and any other items that I reference in my below analysis.

35. I understand that other issues may arise that require further explanation, and I will provide that explanation if appropriate. As a result, I respectfully reserve the right to update and supplement this declaration and the information and opinions provided herein.

VI. PERSON OF ORDINARY SKILL IN THE ART

A. A POSITA's Ordinary Level of Skill

36. I understand that a patent must be analyzed from the perspective of a person of ordinary skill in the art ("POSITA") as of the time of invention of the patent, which is typically considered to be the patent's earliest filing date. Based on my knowledge and experience in the field and my review of the '621 Patent and its file history, I believe that a POSITA would have included a person with:

- (1) an undergraduate degree in electrical engineering, computer engineering, computer science, or comparable field of study, and (2) at least two years' experience working with and programming networked computer systems.

The relevant prior art, including Stringfellow and Kloster, also evidence this level of ordinary skill. The background technology described in Section VII and the prior art described in Sections IX and X demonstrate that a POSITA would have been familiar with the underlying principles of secure transactions, including e-commerce transactions over the Web, Internet, as well as network communication, data transfer, and content sharing across networks, including using the HTTP/HTTPS and TCP/IP protocols. Individuals with different education and additional experience could still be of ordinary skill in the art if that experience compensates for a deficit in their education stated above. Additional education might substitute for some experience, and vice versa.

VII. A POSITA'S UNDERSTANDING OF THE FIELD OF TECHNOLOGY

A. HTTP Communications

1. HTTP Communications

37. Various methods of web communications were well-known by 2012. For example, RFC 2616, published by the Internet Engineering Task Force (IETF), defines HTTP 1.1, which is a protocol facilitating communication between clients and servers over the internet. EX1011, 1/176. HTTP is primarily a request-response protocol where a client, typically a web browser, initiates a request to a web server, often through an HTTP GET message. EX1011, 12/176.

38. RFC 1738, another IETF publication, defines the syntax and structure of a URL (Uniform Resource Locator). EX1009, 1/25. A URL consists of several parts, including the scheme (like “http” or “https”), the server’s hostname or IP address, and the path to the resource. EX1009, 9/25. Additionally, URLs can contain a query string, starting with a “?” symbol, that includes parameters providing further instructions to the server. *Id.* Query strings are generic key-value pairs that can be used for any number of purposes. For example, they may be used to pass session-specific information about a client or desired destination page within a larger server. Or, they may be used to track a user, a session, or data related to either between multiple HTTP requests.

2. *Conveying State/Context Information In Stateless HTTP*

39. HTTP is a stateless protocol, which means that an HTTP server treats each request it receives independently without knowledge of any past history. Accordingly, it was known that if the server needed to be aware of a specific context or relation of a request to an ongoing operation, that must be conveyed to the server. Then, based on information provided by the user in the form of a parameter, user-stored cookie, etc., the server could respond by producing a certain page, or a generating the requested page with session-specific data. By 2012, it was well known to convey such a context or relation to an ongoing operation by including information in the query string of a URL in an HTTP request. For example, Carlson discloses a URL “src=http://www.ivisa.com/zifpvisa/php?Merchid=2000Scorrid=123” that includes “transaction information” in the form of a “merchant ID and “correlation ID” in the URL query. EX1018, 14:42-47, 6:45-49. This information enables the payment processing network receiving the URL to understand that this HTTP request is linked to earlier messages from a “merchant server” with the same merchant ID and correlation ID. *See id.*; *see also* EX1013, 9:21-30 (URL including in query string “information as may be needed to sell an item”); EX1023, 5:32-43. In this way, for example, a server could manage different requests, across different sessions, using a single URL with embedded, session-specific parameters. By 2012, this technique

was well known in the art of computer networking more broadly, as it was valuable to any server managing HTTP requests.

3. *HTML and iFrames*

40. Inline frames (iFrames)—introduced in HTML 4.01 (1999)—allow a web page to display content from one server while also loading separate, isolated content from another server in an embedded iFrame. EX1022, 12/12 (“Inserting an inline frame . . . allow[s] you to insert an HTML document in the middle of another”). Specifically, the iFrame displayed what appeared as a single page to a user where a component of that page within the transparent iFrame was obtained through a different HTTP request—potentially from a different server—and displayed without replacing other portions of the page. *See* EX1007, [0075]; EX1008, [0016].

41. It was also known to populate a state-specific iFrame by using a URL with state information contained in the URL query string parameters. EX1018, 14:42-47 (“<iframe ‘src=‘http://www.ivisa.com/zifpvisa/php?Merchid=2000Scorrid=123 ...”). For example, Boyd discloses a URL to populate an iFrame that includes “information as may be needed to sell an item, such as the seller identification (in our example an email address).” EX1013, 9:21-30.

```
<iframe
src="http://labs.arch.ebay.com/j2ee/GClick/GClickPublicIFrame.html
?email=alogrin@yahoo.com
&pict=http://g.fool.com/art/seminars/cs_getouthead.gif
&file=http://g.fool.com/art/seminars/downloads/debtworkbook.pdf"
height="170" width="250" frameborder="NO"
scrolling="no"></iframe>
```

Id. In the above example, the iFrame's content will be determined by the "src" URL, whereas the rest of the webpage may be loaded from a different server.

4. HTTPS / Secure Socket Layer ("SSL") Connections

42. RFC 6101, published in August 2011, memorializes SSL 3.0 as released in 1996. EX1012, 1/67 (RFC 6101). By 2011, SSL 3.0 was "a widely implemented protocol." *Id.* "The SSL protocol provides connection security that has three basic properties:" (1) "The connection is private" based on use of "encryption . . . after an initial handshake," (2) "The peer's identity can be authenticated using asymmetric, or public key, cryptography," and (3) "[t]he connection is reliable." *Id.*, 5/67.

5. Application Programming Interface (API)

43. In addition to serving requests from web browsers, it was also known for servers, including web servers, to provide services to other software programs, including other servers. See e.g., EX1007, [0067]; EX1008, [0015]; EX1015, [8:65-

9:3]. An API is a function that a server can provide that allows for “integration between the system [server] and client applications” by providing “[c]onfiguration,” “set up,” and/or “functionality” functions to the client. EX1015, 8:65-9:3. It was well known by 2012 for secure financial servers to provide API functionality to other eCommerce servers, such as merchant servers. *See e.g.*, EX1007, [0072]-[0073]; EX1008, [0015]; EX1014, 111:61-63. For example, Raleigh teaches the use of a “network transaction server” that “seeks an API connection with the device mobile payment agent 4699, which then confirms the API connection.” EX1014, 111:61-63. Petri also discloses the use of an “ecommerce solution” that includes “application programming interfaces (API) configured as web services for monetizing in-application activities such as user account management, fraud detection, wallets, and transaction and payment processing.” EX1015, 4:1-8. It was also known to include as part of an API request, information and procedures for authorizing/authenticating the API requestor (i.e., client) to “protect[] sensitive data from unauthorized access.” EX1028, [0007], [0042]; EX1027, [0058]-[0059]; EX1029, 12:61-13:1. This included “requiring the mutual exchange and verification of certificates between the first client and the first server,” as well as including the API request information specific to a requester. EX1028, [0007], [0042] (“Authorization based on API Keys can help differentiate permissions between multiple Clients.”). “API Keys” may include “a random string that is known to both the Client and the” API server and an

“Application ID” identifying the requesting party. EX1028, [0042]; EX1027, [0059].

B. Online Shopping and Payments

1. Online Shopping and Payment Processing

44. By 2012, eCommerce was well known, as was online payments. EX1007, [0003] (“Users are increasingly encouraged to purchase goods online”). Being online, eCommerce communications between merchants, users, and third parties were usually accomplished using standard Internet protocols, such as HTTP. *See e.g.*, EX1007, [0072]; EX1015, 26:36-38; EX1013, 7:49-60; EX1019, 6:20-22. Recognizing a need for added security, these communications typically used HTTPS, an encrypted form of HTTP that combines HTTP with a Secure Socket Layer (SSL) connection. *See e.g.*, EX1007, [0074]; EX1015, 26:36-38; EX1019, 6:10-18; *see also* EX1012, 1/67.

45. Server hosts that receive—and store—sensitive payment information (even over HTTPS) must invest “time and money” to ensure their systems and policies are “in compliance with the credit card networks as defined by various standards such as the PCI Security Standard.” EX1021 1:45-60; *see* EX1001, 1:45-48. As a result, it was also known for eCommerce sites to turn to another provider to collect (and safeguard) payment information. For example, a merchant site could “redirect[] the buyer to an alternative payment system website.” EX1007, [0013];

see also EX1020, 1:25-28; EX1021, 1:33-35. But, this approach results in a “poor user experience,” “causing confusion” to users who are redirected multiple times during payment. EX1021, 1:37-41; EX1008, [0003] (“This is not a seamless process for the end user”).

46. To provide a more seamless experience to the user, it was also well known for eCommerce sites used iFrames to seamlessly embed the user’s HTTPS communications with a secure third-party payment server within the eCommerce site’s own webpage. EX1007, [0075]; EX1008, [0016]; EX1020, 8:52-58. That is, by 2012, it was known to use iFrames to enable a single eCommerce page in which shopping information was displayed based on content sent from one server and payment information was collected via separate instructions rendered within an iFrame sent from a second server. For example, Stringfellow explains “the sign-in page is implemented as an iFrame, which enables the user to communicate directly with the trusted intermediary system 10 while remaining within the online merchant's online environment.” EX1007, [0075]. Likewise, Carlson explains: “The use of IFrames allows the device collection to work transparently and to not be compromised by browser controls, such as pop up blockers”). EX1018, 12:59-61; *see also* EX1020, 3:51-55; EX1019, 2:28-60.

2. Tokenization

47. By 2012, it was well known that if eCommerce servers received secure information, they would be subject to expensive compliance requirements unless tokenization techniques were implemented. *See infra* ¶45. Tokenization replaces sensitive data, such as credit card numbers, with a unique, non-sensitive token that has no exploitable value on its own. EX1021, 1:62-67 (“These gateways offer tokenization technology whereby a merchant can submit sensitive payment information to the gateway and receive a token that serves as a proxy for the sensitive payment data and can be securely stored in the merchant’s systems to execute subsequent charges”). When applied to buyer transactions, a payment server generates a token representing the buyer’s sensitive information and provides it to the merchant so the merchant can process transactions without directly handling or storing sensitive data, as the token can be used with the payment server to complete the transaction. EX1008, [0016] (“the payment system server is able to receive sensitive payment data from an end user without requiring the business entity to collect the sensitive payment data.”). This approach enhanced security by reducing the risk of sensitive data exposure while allowing merchants to handle transactions smoothly. Since the merchant and the payment server used the non-sensitive token as a proxy for a payer’s sensitive information, the merchant server was never privy to the sensitive info. By shielding itself from this information, the merchant server

reduced the risk that sensitive data would be exposed to untrusted third parties. The '621 Patent also acknowledges the well-known use of tokenization where, “credit card processing organizations have used tokenization to provide a link to credit card data for purposes of storage and retrieval.” EX1001, 1:50-52.

VIII. OVERVIEW OF THE '621 PATENT

48. The '621 Patent relates to “systems and methods for obtaining and using account information to process financial payments.” EX1001, 1:18-20. The '621 patent describes a tokenization process in which a “[m]erchant server 204 may store customer identification information, purchase history, and other data that does not include financial account information such as credit card information, without being subject to PCI compliance requirements.” *Id.*, 7:8-12; *see also id.* 5:62-6:4. A “[s]ecure server 202...stores the financial account information received from the customer.” *Id.*, 7:31-34.

49. The merchant server initiates two separate and distinct functions during the payment process: the financial account registration function and the token retrieval function. *Id.*, 5:57-62, 6:63-7:1. First, in response to a payer’s request to make a purchase from the merchant, the merchant server initiates the financial account registration function, which instructs the secure server to establish a communication link with the payer. Through that communication link, the payer can submit her sensitive financial account information to the secure server to be stored

and tokenized. *Id.*, 7:66-8:2. The secure server then “communicates with the payer” so that the “payer is preferably able to provide the sensitive financial account information to the [secure server] without this information being visible or audible to personnel employed by the operator of” the merchant server. *Id.*, 8:26-29, 8:32-36. To generate a communication link between the payer and the secure server, the “[m]erchant server 204 may provide the customer at site 206, 208 or 210 with a widget or redirected URL that provides an online window for direct, preferably secure communication with the secure server 202 for entry of financial account information.” *Id.*, 9:55-59.

50. Second, the secure server stores the sensitive financial account information and “a token is generated for the referring payee (e.g. the merchant or operator of the merchant server).” *Id.*, 8:37-40. “The token is a pointer that allows the payee to reference the financial account information of that payer without possessing the actual financial account information.” *Id.*, 8:40-43.

51. Third, to process the payment from the payer, the merchant then initiates the token retrieval function. The merchant server “receives the token from the” secure server and “[t]he token is stored in the [merchant server] in association with the electronic enrollment data for the payer.” *Id.*, 8:44-48. Then, “merchant server 204 submits the token corresponding to the customer account to be debited and information specifying the amount of the payment to the secure server 202.” *Id.*,

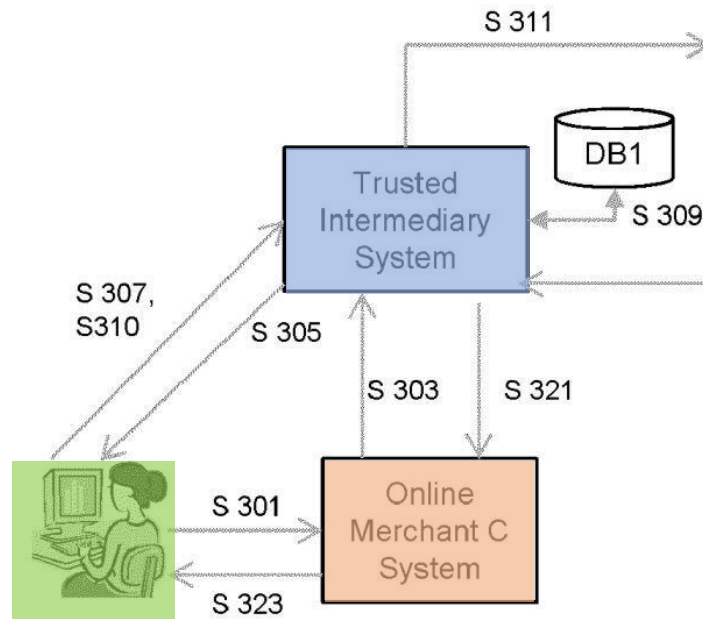
9:12-16. “The operator of the secure server 202 then processes the payment using the financial account pointed to by the token.” *Id.*, 9:16-19.

52. However, as I will explain with reference to Stringfellow, Kloster, and the background knowledge of a POSITA, the ’621 Patent merely describes standard payment systems and methods of securing communications of financial information within these systems.

IX. GROUND 1: STRINGFELLOW AND KLOSTER

A. U.S. Patent Application Pub. No. 2010/0175626 A1 (“Stringfellow”)

53. Stringfellow discloses “a method of processing payment authorization requests for payment transactions.” EX1007, Abstract. The “user” (green, meets the claimed “payer computer system”) is served by a first server, “online merchant system” (red, meets the claimed “merchant server”) providing an online shopping experience and a second server, “trusted intermediary system” (blue, meets the claimed “one or more secure servers”), providing payment authorization functionality, including registration and collection and storage of sensitive financial information, “on behalf of online merchants.” *Id.*, Abstract, FIG. 3.

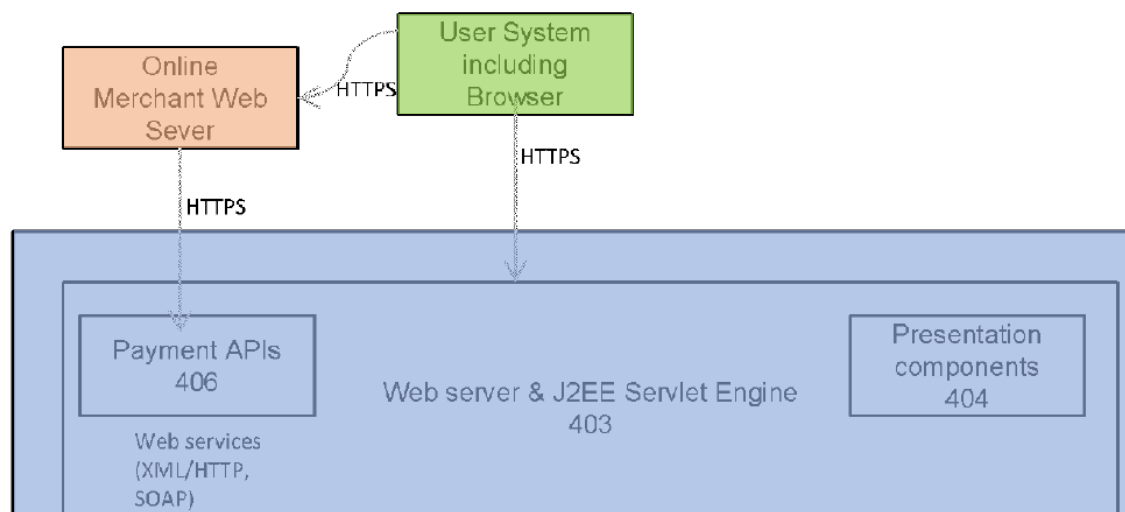


Id., FIG. 3 (excerpt, annotated).

54. The “sign-in page” served by the trusted intermediary system “is implemented as an iFrame” within a page served by the online merchant system, “which enables the user to communicate directly with the trusted intermediary system 10 while remaining within the online merchant’s online environment.” EX1007, [0075]. Stringfellow describes the message flow used to accomplish this user experience “with reference to FIG. 3” and then describes those same steps “in more detail” with reference to “FIG. 5.” EX1007, [0075]. Accordingly, a POSITA would understand these disclosures describe a common embodiment.

55. Stringfellow also explains the trusted intermediary system can communicate directly with the online merchant system via “web server and servlet engine 403” that includes “web services-based payment APIs or API wrappers to

online merchant systems.” EX1007, [0072]-[0073], [0076]. Further, the trusted intermediary system receives “authorization request[s]...by the payment APIs 406” for processing payment transactions. EX1007, [0076]. Thus, Stringfellow discloses the trusted intermediary system providing an application programming interface (API) to the merchant server for serving API requests from the online merchant server.



EX1007, FIG. 2 (excerpted, annotated).

1. Online Merchant System → Trusted Intermediary: Originating Payment Authorisation Request Message.

56. As part of an online shopping experience hosted by the online merchant system, the online merchant system presents code that enables the user to “select[s] ‘Secure System for Payment’ (SSP) as a payment option” (step S301 / S5.1). EX1007, [0045], [0075]. Upon receipt of such a selection from the user, “the online

merchant system [] transmit[s] an originating payment authorisation request message to the trusted intermediary system” (step S303 / S5.3). EX1007, [0045], [0075]. This “request message” includes “at least an amount of payment for the selected goods, the online merchant account identifier and an identifier for the order.” EX1007, [0045].

2. Trusted Intermediary → Online Merchant System: Generation and Retrieval Of Sign-In Page URL.

57. The originating payment authorisation request message also requests “retrieval by the online merchant system of the URL corresponding to a sign-in page of the trusted intermediary system” for “display[]...to the user.” EX1007, [0075].

58. In connection with Figure 3, Stringfellow states the “trusted intermediary system” “transmits a login URL to the user.” EX1007, [0045]. In view of the “more detail” of this step described in connection with Figure 5, I would understand this passage refers to the online merchant system retrieving the URL and transmitting instructions to the client’s browser to retrieve the URL.

3. Online Merchant System → Client: Provide Sign-In Page URL Via iFrame; Client ↔ Trusted Intermediary: Request URL, Establish SSL Session, Serve Sign-In Page.

59. “Having received the sign-in URL,” “the online merchant system displays the sign-in page to the user” (step S305 / S5.5). EX1007, [0045], [0075]. “[C]ertain embodiments make use of iFrame web technology to navigate the user to different websites,” including “[i]n one arrangement the sign-in page is implemented

as an iFrame, which enables the user to communicate directly with the trusted intermediary system 10 while remaining within the online merchant's online environment." EX1007, [0077], [0075]. A POSITA would understand that, based on web principles, this is accomplished by: (a) the online merchant system sending to the client browser code instructing the browser to send an HTTP request to the trusted intermediary system specifying the sign-in URL, and (b) the trusted intermediary system sending an HTTP response to the client browser with instructions to display the sign-in page. It was common in the art for websites to use iFrames to embed content retrieved from other servers within the visual interface of their own webpage. Accordingly, the iFrame code would contain a URL to the other server and client browsers would interpret this code as meaning they should make a HTTP request to this URL. In the case of Stringfellow, this means that the online merchant system's iFrame would contain the URL for the sign-in page of the trusted intermediary system and the client browser would send an HTTP request to retrieve the sign-in page based on the URL in the merchant system's iFrame.

60. "When implemented as a web server, data between the elements of the payment system 1 (i.e. those shown in FIGS. 2 and 3)," which include the client browser, "and the trusted intermediary system 10 are transmitted using a secure mechanism, e.g. via the HTTP over Secure Socket Layer protocol (HTTPS)." EX1007, [0074]. Therefore, the sign-in URL would be an HTTPS request and, prior

to communicating the sign-in page, the user's browser and the trusted intermediary server necessarily establish an SSL session (secure session). It was well known in the art that HTTP and SSL can be layered on top of each other to form an HTTPS connection that is encrypted using SSL. It would have been understood that all of the connections in the payment system should be made over HTTPS for security reasons, especially the link between the user and the trusted intermediary, which will be used to transmit the user's payment information.

4. Client $\leftrightarrow$ Trusted Intermediary: Account Registration, Including Collecting and Storing Sensitive Financial Information.

61. The sign-in page "prompt[s] the user to login, or, if this is their first time of selecting SSP as a payment option, to register with the trusted intermediary system." EX1007, [0045], [0075]. If the user seeks to register, the registration flow may be accomplished via "[r]e-direct from order system" where the "user is re-directed to the registration screens associated with the trusted intermediary system 10 and then re-directed back to the online merchant's system." *Id.*, [0055]. In the embodiment in which the sign-in page is populated in an iframe, A POSITA would understand that the trusted intermediary system provides the registration screens for display within that same iframe. As explained above, an iFrame can be used as a container for another server's URL. In this case, the trusted intermediary's sign-in page URL would be placed in the iFrame, the user's browser would make a HTTP

request to the sign-in page URL when it loads the iFrame, and then the response to that request (the sign-in page itself) would populate the area of the iFrame that is visible to the user. After that point, the user can continue to interact with the various registration screens associated with the trusted intermediary within the iFrame, the trusted intermediary's webpage controlling the HTTP redirects within the iFrame.

62. During this registration, the “account is required to be populated with appropriate data that can be used to make payments from the SSP service.” EX1007, [0051]. “Once registered, and as mentioned above each user has a set of records associated therewith, which stores details of the accounts which they wish to debit when effecting a financial transaction,” which may include “a bank account, a payment card or other account such as any payment instrument that can be given a unique account reference.” *Id.*, [0052]. A POSITA would understand that this is accomplished by the trusted intermediary sending instructions to the client's browser to render a form for the user to enter sensitive financial information and, upon completion, the client's browser sends the information over the SSL connection to the trusted intermediary and stored at the trusted intermediary. It was well known for webpages to provide forms to users to receive user input, including to receive payment information. A POSITA would understand Stringfellow's teaching that the account be “populated” with the user's financial information to mean that the trusted

intermediary's registration screens would ask for this information using one or more forms.

63. Upon completion of the registration flow, the user's browser is "re-directed back to the online merchant's system." EX1007, [0055].

64. Stringfellow also describes an alternative flow where the "user has previously registered with the service" in which the system presents a screen for "selection of payment method" and, upon selection, makes a request to the "merchant IPSP system" to complete the transaction. EX1007, [0045]-[0047].

5. *Content of "Sign-In URL."*

65. A POSITA would understand that the sign-in URL that the trusted intermediary system provides to the online merchant system in response to an originating payment authorisation request message includes information linking the URL to that request message. This is necessarily true because Stringfellow describes the trusted intermediary, in response to the user logging in and selecting a method of payment, effecting payment by contacting a back end system (the "merchant IPSP system") with "the amount of payment required and the online merchant identifier." EX1007, [0046]. But the trusted intermediary system does not obtain the amount of payment or online merchant identifier via the information the user submits. *Id.* Accordingly, the trusted intermediary system must be able to associate the request for the sign-in URL with the corresponding transaction started by the merchant

server and the corresponding submitted information (e.g., “at least an amount of payment for the selected goods, the online merchant account identifier and an identifier for the order”). A POSITA would understand that the sign-in URL must, therefore, include an identifier sufficient to tie the sign-in URL to a particular transaction. EX1007, [0045]. Further, as explained, it was well known to modify URLs with query string parameters or otherwise generate custom URLs that correspond to a specific session. A POSITA would understand that the sign-in URL must be tailored to the user to allow the trusted intermediary system to identify the user when the user makes a request to the URL provided in the merchant system’s iFrame.

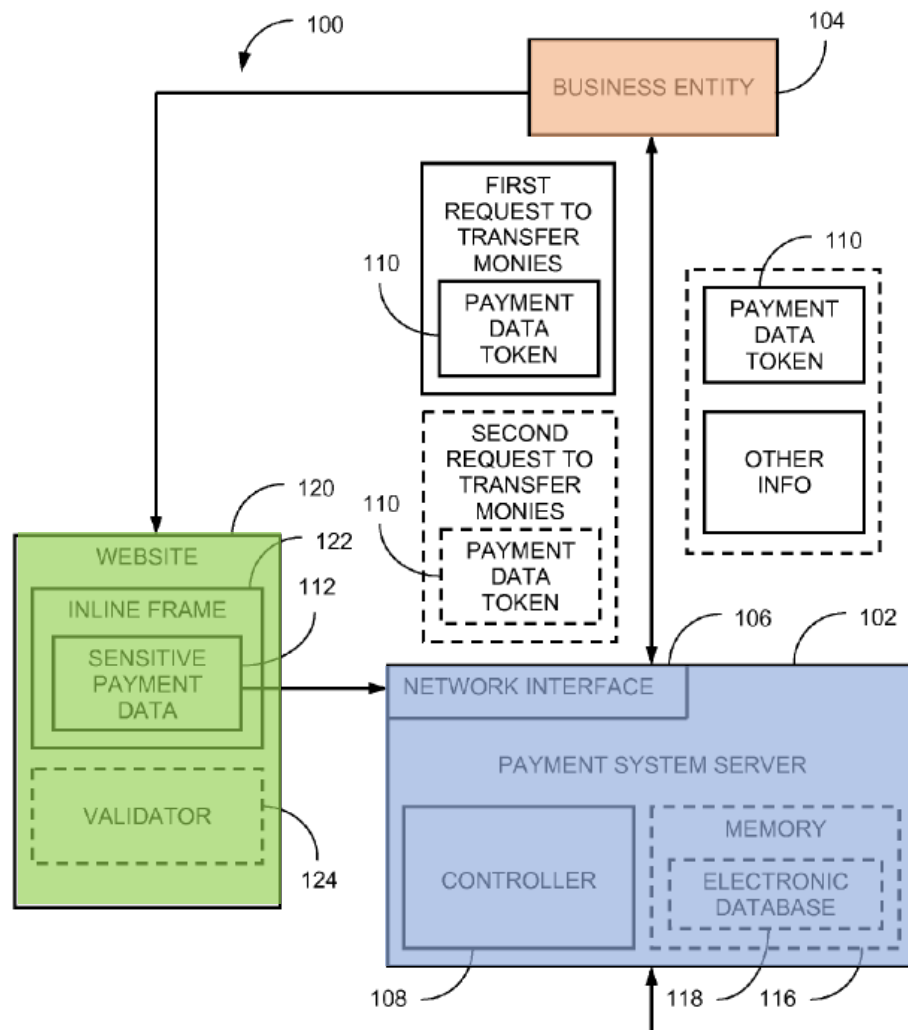
66. As discussed in Sections VII.A.2-VII.A.3, it was well known to tie a new HTTP message to an ongoing transaction by including information in the query string of a URL in an HTTP request that identifies the transaction or information from the transaction. Accordingly, Stringfellow requires that, in returning a sign-in URL to the online merchant server, the trusted intermediary server generated the URL to include information in the query string tying that URL to a transaction and, therefore the information for that transaction submitted with the originating payment authorization request message. As evidence that this approach would have been obvious to a POSITA, Carlson, discloses a merchant server submitting a “correlation ID” to a secure server that “ties together messages associated with the same

transaction” and then including that correlation ID, along with a merchant ID, in the query string of a URL sent to the secure server to populate an iframe of a checkout page: “<iframe src= ‘http://www.ivisa.com/zifpvisa/php?Merchid=2000Scorrid=123 ...>.” EX1018, [0080], [0100]. Stringfellow discloses the online merchant system providing two transaction identifiers to the trusted intermediary in the original request message: “the online merchant account identifier and an identifier for the order.” EX1007, [0045]. As explained in the background of HTTP technology, passing messages, data, and other information using the URL query string parameters was well known in the art. It would, thus, be obvious to include those same two transaction identifiers in the query string of the sign-in URL to tie together these messages associated with the same transaction. EX1018, [0080], [0100].

B. U.S. Patent Application Pub. No. 2010/0094755 A1 (“Kloster”)

67. Kloster, like Stringfellow, discloses a method “for collecting sensitive payment data” by presenting a single web page hosted by a customer. As annotated below in Kloster’s Figure 1, a [green] “web page 120” (served by the customer’s computer, *i.e.*, the claimed “payer computer system”), where a first server, [red] “business entity 104” (*i.e.*, the claimed “merchant server”), provides content for an online shopping experience and a second server, [blue] “payment system server 102” (*i.e.*, the claimed “one or more secure servers”), provides functionality for

“collecting sensitive payment data” hosted within an “iFrame [] embedded directly within a website maintained by a business entity.” EX1008, Abstract, [0016], FIG. 1. “By embedding the inline frame 122 directly within the website 120 of the business entity 104, the payment system server 102 is able to receive sensitive payment data 112 from an end user without requiring the business entity 104 to collect the sensitive payment data 112.” *Id.*, [0016]; *see also id.*, Abstract.



Id., Fig. 1(excerpted, annotated). Also like Stringfellow, the “payment system server 102 includes a network interface 106 for communicating with the business entity 104.” *Id.*, [0012].

68. Kloster also discloses the use of “payment data tokens” to enable the “business entity” to make requests to the “payment system server” for payments without possessing “sensitive payment data.” EX1008, Abstract. “By providing the business entity 104 with a payment data token 110, the payment system server 102 may provide an online merchant with one-time payment processing, customer data profile management, and/or recurring payment options without requiring the merchant to collect or store sensitive information, such as credit card numbers, or the like.” EX1008, [0021].

C. Motivation to Combine Stringfellow and Kloster

69. A POSITA would be motivated to: (a) add Kloster’s disclosure of the use of “payment data tokens” by its “business entity 104” and “payment system server 102” to the functionality of Stringfellow’s “online merchant system” and “trusted intermediary system,” respectively, to obtain the benefits Kloster explicit describes, and (b) utilize additional implementation details Kloster provides for functions that Stringfellow also describes but with less detail. As discussed in Sections **Error! Reference source not found..A-Error! Reference source not found..B**, Stringfellow and Kloster disclose the same architecture to achieve the

same task—enabling an “online merchant system” / “business entity” to rely on the “trusted intermediary” / “payment system server” to obtain sensitive payment information directly from a user and achieve payment. Thus, a POSITA would look to Kloster both as providing additional compatible functionality and additional implementation details for implementing common functionality.

70. Further, both Stringfellow and Kloster recognize the advantage of this arrangement of avoiding the risks and costs associated with the “online merchant system” / “business entity” obtaining and storing sensitive payment information. EX1007, [0012] (“for each new online merchant that a buyer interacts with, the risk of exposure ... of the buyer’s financial data increases”), [0028], [0039] (disclosed architecture “has the benefit of reducing the risk of fraud that may be incurred in relation to conventional arrangements”); EX1008, [0003] (“the online merchant may be exposed to financial risk in the event of a security breach”), [0016] (disclosed architecture means “the payment system server 102 is able to receive sensitive payment data 112 from an end user without requiring the business entity 104 to collect the sensitive payment data”).

71. Kloster explains that a system with that common architecture can obtain the benefit of additional functionality while still retaining the benefit of the merchant system not being required to store sensitive information through the use of tokens: “By providing the business entity 104 with a payment data token 110, the payment

system server 102 may provide an online merchant with one-time payment processing, customer data profile management, and/or recurring payment options without requiring the merchant to collect or store sensitive information, such as credit card numbers, or the like.” EX1008, [0021]. Thus, a POSITA would be motivated to include Kloster’s token functionality in Stringfellow to enable Stringfellow’s “merchant server” to perform additional functionality including “one-time payment processing” via a direct request from the “merchant server” to the “trusted intermediary” and “recurring payment options” for subsequent transactions while maintaining the basic premise of both Stringfellow and Kloster of the “merchant server” not storing sensitive financial information. It is likely that users will want to make repeat purchases from the same merchant. Storing a token identifying the user without storing sensitive information, would only augment the convenience of Stringfellow’s payment system for the user.

72. As one specific example, a POSITA would have been motivated to add Kloster’s “payment data tokens” to Kloster to add support for collection of a second payment without requiring the user to provide financial information to either the merchant server or the trusted intermediary server again—beneficial functionality Stringfellow does not disclose. Kloster explains that “if the online merchant would need to collect a payment again utilizing the same payment data (e.g., to renew a monthly subscription), the merchant may do so utilizing the same payment data

token.” EX1008, [0026]. This eliminates the need for the customer to repeatedly provide their payment details, thus reducing the risk associated with transmitting and storing sensitive data.

73. Kloster also describes additionally benefits of use of tokenization, including “validat[ing] information regarding the transaction prior to processing the transaction.” EX1008, [0025]. By relying on the tokenization functionality provided by the secure payment system server as disclosed in Kloster, the merchant can avoid handling sensitive data directly while still being able to access secure services, such as fraud detection algorithms and identity verification mechanisms, without needing to build or maintain such systems in-house.

D. Modification of Stringfellow in view of Kloster

74. Applying Kloster’s teachings to Stringfellow results in: (a) incorporating the functionality for requesting and using tokens of Kloster’s “business entity” into Stringfellow’s corresponding “online merchant system” and incorporating the functionality for generating, providing, and responding to requests including tokens of Kloster’s “payment system server” into Stringfellow’s corresponding “trusted intermediary system,” and (b) utilizing Kloster’s additional disclosures of compliance with security standards in storing sensitive financial information. Stringfellow and Kloster share a common architecture and, thus, the correspondence between Kloster’s “business entity” and “payment system server”

with Stringfellow’s “online merchant system” and “trusted intermediary system,” respectively, enables straightforward incorporation of this functionality. Both references operate using the same exemplary eCommerce network: a consumer, a merchant, and a third-party trusted payment server that manages the payment by interfacing with both the consumer and the merchant. A POSITA would understand that the two systems could be combined as described herein.

75. With those functions integrated, tokens can advantageously be used in multiple payment flows within Stringfellow. First, as discussed in Section **Error! Reference source not found.**A, Stringfellow discloses displaying registration screens in an iframe during which the “account is required to be populated with appropriate data that can be used to make payments” and, upon completion, the user’s browser is “re-directed back to the online merchant’s system.” EX1007, [0051], [0055]. Kloster likewise discloses “sensitive payment data” “is collected via the inline frame” and further explains that at the conclusion of that process, “a payment data token is associated with the sensitive payment data” and the “payment data token is provided to the business entity.” EX1008, [0023]. Thus, starting from the common step of collecting sensitive payment information via an iFrame, Stringfellow’s registration flow would continue in accordance with Kloster’s disclosure: creating and providing a payment token to the merchant server, the

merchant server submitting a payment request using the token, and the secure server processing payment.

76. Applying Kloster's disclosure, Stringfellow's online merchant system may obtain a payment token after "sensitive payment data" "is collected" in multiple ways. "[I]n a specific implementation," "the inline frame may be redirected to a call back URL specified by the merchant," here Stringfellow's online merchant system, such that "the payment data token is passed as a parameter to the call back URL." EX1008, [0023]. Alternatively, Kloster discloses "[i]t is contemplated that any non sensitive data could be retrieved via an inline frame *or API calls by the merchant.*" EX1008, [0015]. As discussed in Section **Error! Reference source not found..A**, Stringfellow discloses the online merchant system sending API requests to the "web services-based payment APIs" of "web server and servlet engine 403" of the trusted intermediary system. EX1007, [0072]-[0073], [0076]. Thus, the online merchant system may request the payment token via that API.

77. After receiving the payment token, the merchant, here Stringfellow's online merchant server, then sends a "request" to the payment system server, here Stringfellow's trusted intermediary, "to transfer at least a portion of the monies" and includes the "payment data token" with the request. EX1008, [0024]. Again, the "web services-based payment APIs" provides an established path for this message. EX1007, [0072]-[0073], [0076]. Finally, Kloster discloses "the sensitive payment

data regarding the access to the monies held for the end user is retrieved from the electronic database” and “a portion of the monies held for the end user is transferred to the business entity utilizing the retrieved sensitive payment data via a backend payment gateway.” EX1008, [0024]. This functionality would be implemented using Stringfellow’s parallel architecture in which the sensitive payment data stored in “DB1” is retrieved by the trusted intermediary for use in requests sent to a backend payment gateway, “the merchant IPSP system.”

78. In a second payment flow, Kloster states “if the online merchant would need to collect a payment again utilizing the same payment data (e.g., to renew a monthly Subscription), the merchant may do so utilizing the same payment data token” through another payment request. EX1008, [0026]. This second payment request would use the same API in Stringfellow and follow the same steps as discussed above.

E. Element-by-Element Analysis

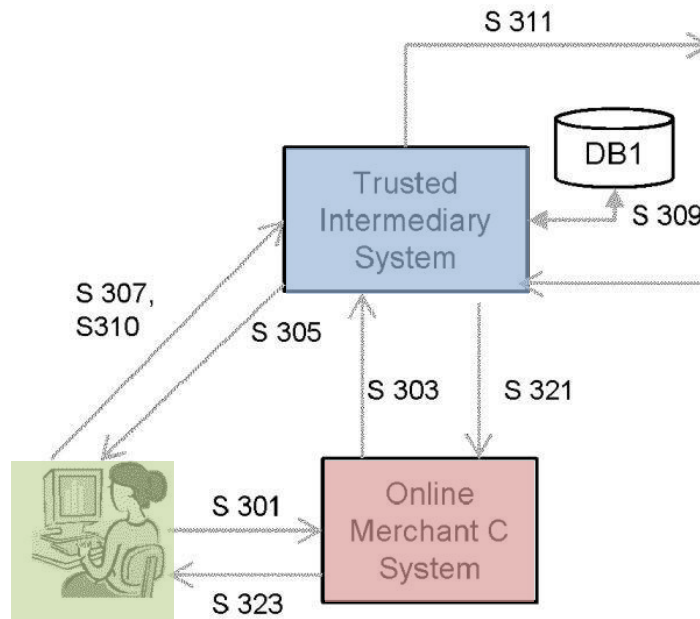
79. The analysis of each element below incorporates the above descriptions of each reference, the motivation to combine the references, and the resulting modification.

1. Claim 1

- a) [1.Pre] A method of processing a payment transaction from a payer to a payee, the method being performed by one or more secure servers, the method comprising:**

80. In my opinion, Stringfellow/Kloster renders obvious Limitation [1.pre].

As discussed in IX.A and annotated below in Stringfellow's Figure 3, Stringfellow discloses a method for processing payment authorization requests for payment transactions from a [green] user that is the "payer" to a [red] online merchant system that is the "payee." EX1007, Abstract, Fig. 3, 5, [0045]-[0047], [0075]-[0076]. The [blue] trusted intermediary system is the "one more secure servers" that performs the method below via a data communications network on behalf of the online merchant system. *Id.* The trusted intermediary system is "secure" because "users only have to submit their respective payment details, preferably only once, to a separate, trusted entity" that securely collects and stores that payment information. EX1007, [0028], [0039]; EX1008, [0003], [0028], [0039].

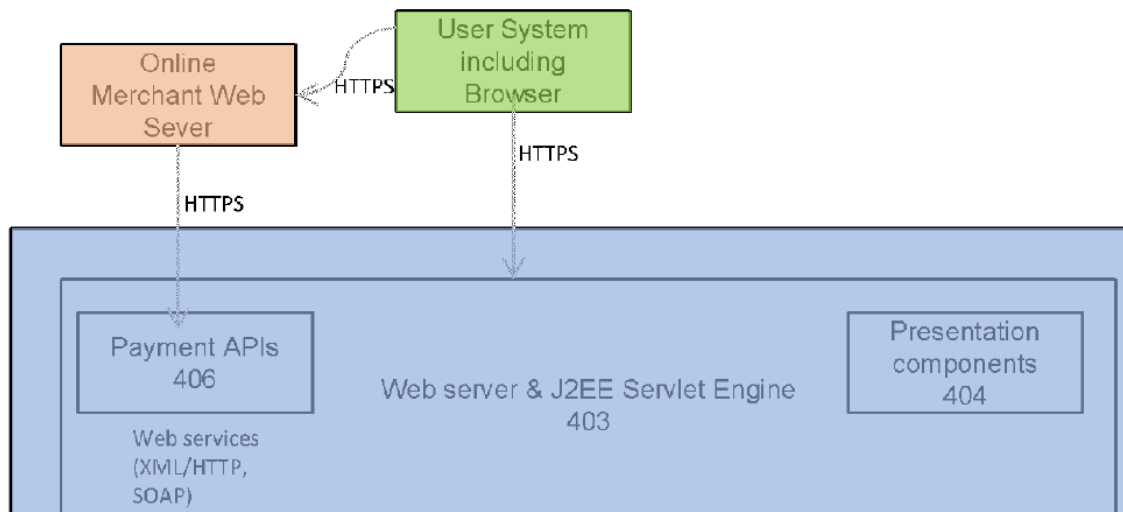


EX1007, Fig. 3 (excerpt, annotated).

- b) **[1.1] providing, by the one or more secure servers to a merchant server providing a webpage to a payer computing system used by the payer, an application programming interface (API) that:**

81. In my opinion, Stringfellow/Kloster renders obvious Limitation [1.1].

As discussed in IX.A, the trusted intermediary system provides an application programming interface (API) through “presentation components, which expose web services-based payment APIs or API wrappers to online merchant systems [in red, above].” EX1007, [0072]-[0073], [0076], Fig. 4. Further, the trusted intermediary system receives “authorization request[s]...by the payment APIs 406” for processing payment transactions. EX1007, [0076].



EX1007, FIG. 2 (excerpted, annotated).

82. Stringfellow also discloses that the online merchant system provides a website where “the user selects SSP payment service as the payment method and submits their selection to the online merchant website.” EX1007, [0075].

c) **[1.1.1] provides financial account registration and token retrieval functions that can be executed to process the payment transaction;**

83. In my opinion, Stringfellow/Kloster renders obvious Limitation [1.1.1]. As discussed in IX.A, the trusted intermediary system provides for financial account registration through its “a method of processing payment authorization requests for payment transactions” “on behalf of online merchants.” EX1007, [0038], [0045], [0075]-[0077], Figs. 3, 5.

84. Further, as discussed in IX.D, the Stringfellow/Kloster combination provides a “payment data token” to the “business entity” that can be used to later

request “transfer at least a portion of the monies held for the end user to the business entity, 212.” EX1008, [0015], [0023]-[0024], Fig. 2; *see also* Limitations [1.1.5]-[1.1.5.4], [1.4]-[1.4.2].

d) [1.1.2] provides access to the financial account registration and token retrieval functions to the merchant server;

85. In my opinion, Stringfellow/Kloster renders obvious Limitation [1.1.2]. As discussed in IX.A and IX.D, the Stringfellow/Kloster combination provides access to the merchant server of the financial account registration and token retrieval functions through its “web services-based payment APIs or API wrappers to online merchant systems.” EX1007, [0072]-[0073], [0076]. API is a broad term in the art that is used to refer to a wide variety of function interfaces made available to allow information, commands, or other data to be sent between servers. In line with one of the uses of the term, a POSITA would understand that the Stringfellow/Kloster combination would provide access through API configurations that allow the merchant server to initiate financial account registration and tokenization functions for processing payment transactions. *See also* Limitation [1.1.1].

e) [1.1.3] receives, from the merchant server via the API, at least one data element associated with the payer and a payment amount from the payer to the payee;

86. In my opinion, Stringfellow/Kloster renders obvious Limitation [1.1.3]. As discussed in IX.A, the trusted intermediary receives from the merchant server a data element in the form of an “originating payment authorisation request message”

that “comprises at least an amount of payment for the selected goods, the online merchant account identifier and an identifier for the order.” EX1007, [0045]. “An identifier for the order” is necessarily associated with the payer that created the order and a POSITA would understand as much. EX1007, [0049]; EX1008, [0015].

f) [1.1.4] authenticates the payee; and

87. In my opinion, Stringfellow/Kloster renders obvious Limitation [1.1.4]. It would be obvious to a POSITA to perform authentication as part of servicing an “originating payment authorisation request message” in view of Stringfellow’s description of the nature of those requests and the nature of the application. As discussed in VII.A.5, it was known to include as part of API request information and procedures for authorizing/authenticating the API requestor to “protect[] sensitive data from unauthorized access.” Given the fact that the online merchant server has the ability to request initiation of a financial transaction, it would be obvious, consistent with Stringfellow’s general goal of “reducing the risk of fraud,” to require an online merchant system to provide authentication information when contacting the trusted intermediary system. Stringfellow describes at least two aspects that the “originating payment authorisation request message” that are consistent with performing authentication. First, Stringfellow discloses that “data between the elements of the payment system 1 (i.e. those shown in FIGS. 2 and 3),” including the online merchant server, “and the trusted intermediary system 10 are transmitted

using a secure mechanism, e.g. via the HTTP over Secure Socket Layer protocol (HTTPS).” EX1007, [0074]. SSL supports three handshake modes, including “authentication of both parties,” i.e., the client (the online merchant system) and the server (the trusted intermediary system), in which “authenticated clients must supply an acceptable certificate” (authentication credentials identifying the payee) “to the server” to be authenticated. EX1012, 61-62/67. Thus, the trusted intermediary server would authenticate the payee (i.e., the merchant associated with the online merchant system) as part of an “authentication of both parties” SSL handshake. It would be obvious to a POSITA to use this “authentication of both parties” mode of the SSL handshake given the limited number of options, the SSL standard’s statements criticizing another of the modes (“completely anonymous” “is vulnerable to man-in-the-middle attacks and is therefore strongly discouraged”), and the nature of the application. Id., 48/67. That is, given the fact that the online merchant server has the ability to request initiation of a financial transaction, it would be obvious, consistent with Stringfellow’s general goal of “reducing the risk of fraud,” to require an online merchant system to provide authentication information when contacting the trusted intermediary system. Second, as discussed in IX.A, Stringfellow discloses the online merchant system sending API requests that include “the online merchant account identifier” as part of the “originating payment authorisation request message” and that this same identifier had been previously stored in an “online merchant profile

and registration data.” EX1007, [0045], [0072]-[0073], [0064], [0076]. Based on these teachings, it would be obvious to a POSITA to use the “online merchant account identifier” as an access credential that is compared with the stored merchant account identifier to authenticate the merchant. EX1007, [0045], [0064]. The use of such a merchant ID to perform authentication is consistent with known API authentication techniques as discussed in VII.A.5.

g) [1.1.5] executes the financial account registration function, upon initiation by the merchant server, by:

88. In my opinion, Stringfellow/Kloster renders obvious Limitation [1.1.5]. As discussed in IX.A, the online merchant system initiates the financial account registration function by transmitting “originating payment authorization request message to the trusted intermediary system (step S303)” where “[t]he trusted intermediary system [] then transmits a login URL to the user (step S305),” as described in Figs. 3 and 5. EX1007, [0045], [0075]-[0077], Figs. 3, 5; *see also* Limitations [1.1.5.1]-[1.1.5.4].

h) [1.1.5.1] generating a uniform resource locator (URL), for establishing a secure socket layer connection via the internet between the secure server and the payer computing system, the URL comprising either:

89. In my opinion, Stringfellow/Kloster renders obvious Limitation [1.1.5.1]. As discussed in IX.A, the trusted intermediary system “transmits a login URL to the user (step S305), prompting the user to login, or, if this is their first time

of selecting SSP as a payment option, to register with the trusted intermediary system.” EX1007, [0045], [0075]. A POSITA would understand that to provide a sign-in URL linked to the request message as discussed in IX.A, the trusted intermediary system must generate that URL. For example, since the content of the request message is not known until it is received, a sign-in URL based on this information can be generated only after the request message is received. After providing the sign-in page URL, the trusted intermediary system “subsequently...[provides for] the creation of a secure session.” EX1007, [0075] (emphasis added). Stringfellow also explains that “data between the elements of the payment system 1 (i.e. those shown in FIGS. 2 and 3) and the trusted intermediary system 10 are transmitted using a secure mechanism, e.g. via the HTTP over Secure Socket Layer protocol (HTTPS).” EX1007, [0074], Fig. 2. The URL provides for the creation of a “secure session” with the user based on an “HTTP over Secure Socket Layer protocol (HTTPS).” HTTPS was known as far back as the 1990s and was in use by 2012. It continues to be a common vehicle for encrypted web communications today. Between background knowledge and the explicit reference to HTTPS made in Stringfellow, a POSITA would have understood in 2012 that a reference to a “secure session” refers to a session over HTTPS. *See also* Limitation [1.1.5.1a].

- i) **[1.1.5.1a] [the URL comprising either] a dynamic URL generated by the secure server for the payer and the payee; or a static URL and a hypertext transport protocol (HTTP) parameter used by the secure server to identify the payer and the payee;**

90. In my opinion, Stringfellow/Kloster renders obvious Limitation [1.1.5.1a]. As explained in IX.A, a POSITA would understand that the sign-in URL generated by the trusted intermediary must include information tying the request for the sign-in URL sent by the payer computer system to a specific transaction and, therefore, it would be obvious to include in the query string of the sign-in URL the two transaction identifiers to the trusted intermediary in the original request message: “the online merchant account identifier and an identifier for the order.” EX1007, [0045], [0075]. The “merchant account identifier” is “for” and “to identify” “the payee” and “the identifier for the order” is “for” and “to identify” at least “the payer” because it is specific to a transaction involving the payer. *See also* Limitation [5.1]. Additionally, Kloster discloses additional transaction identifiers, including “a unique customer identifier created by the merchant.” EX1008, [0015]. It would be obvious to include this additional transaction identifier in the original request message and in the sign-in URL query string to provide further information that ties together messages from the same transaction. Regardless of whether including parameters in a query string of a URL is considered generating a “dynamic” URL or “a static URL and a hypertext transport protocol (HTTP) parameter,” the trusted

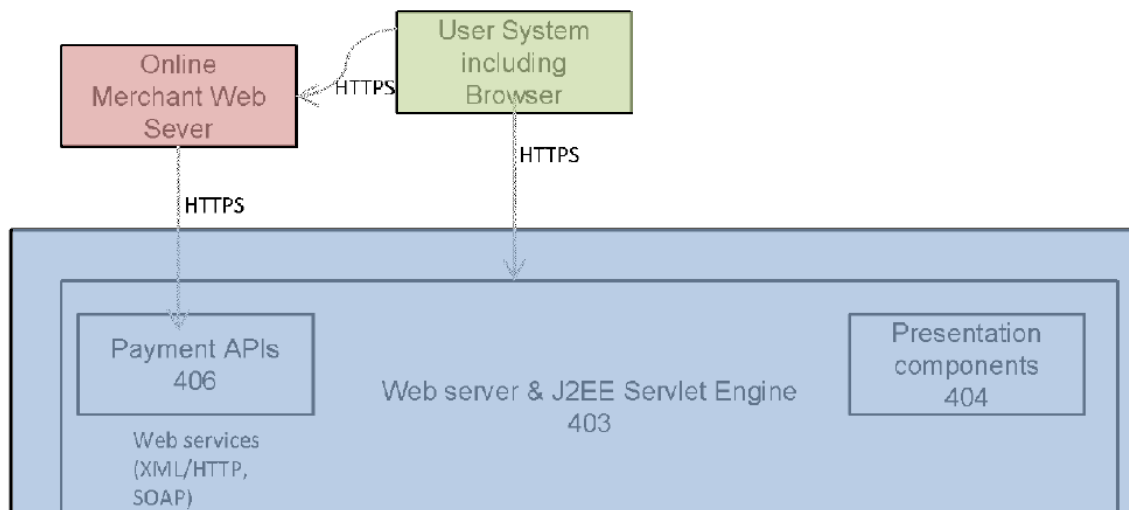
intermediary's generation of the sign-in URL meets at least one of the recited options.

- j) **[1.1.5.2] establishing the secure socket layer connection, in response to an HTTP request received from the merchant server for the generated URL, between the secure server and the payer computing system within a window or frame that is displayed within the webpage provided by the merchant server;**

91. In my opinion, Stringfellow/Kloster renders obvious Limitation [1.1.5.2]. As discussed in IX.A, in response to the merchant server's payment authorisation request message retrieving "the URL corresponding to a sign-in page of the trusted intermediary system," the trusted intermediary system "subsequently [... provides for] the creation of a secure session." EX1007, [0075] (emphasis added). Stringfellow discloses "[h]aving received the sign-in URL from the trusted intermediary system 10, the online merchant system displays the sign-in page to the user" by designating the "sign-in page" to appear "as an iFrame" within the existing online merchant web page. EX1007, [0075]. I understand this to mean that, once the iFrame is displayed to the user, the user's browser detects the HTTPS protocol in the iFrame's src attribute and then initiates the "secure session" through an "initial handshake" with the trusted intermediary system as discussed in VII.A.4. Further, as Stringfellow explains, the "data between the elements of the payment system 1 (i.e. those shown in FIGS. 2 and 3) and the trusted intermediary system 10 are

transmitted using a secure mechanism, e.g. via the HTTP over Secure Socket Layer protocol (HTTPS).” EX1007, [0074], Fig. 2.

92. Further, the merchant server’s “payment authorization request message” must be conveyed as an HTTP request because “the trusted intermediary system [] is preferably embodied as a web application server...which acts as the entry point for external HTTP requests to the trusted intermediary system 10 from online merchants and from users' browsers.” EX1007, [0072] (emphasis added), FIG. 4.



EX1007, FIG. 2 (excerpted, annotated).

- k) **[1.1.5.3] outputting instructions to the payer computing system, in response to the HTTP request for the generated URL, to render a financial account registration request form, within the window or frame that is displayed within the webpage provided by the merchant server, that provides functionality for the payer to provide sensitive financial account information associated with a financial account; and**

93. In my opinion, Stringfellow/Kloster renders obvious Limitation [1.1.5.3]. As discussed in IX.A, in response to the merchant server's payment authorization request message retrieving the sign-in URL, the iFrame displays the URL, prompting "the user to login, or, if this is their first time of selecting SSP as a payment option, to register with the trusted intermediary system 10." EX1007, [0045]. As Stringfellow explains, when registering, the user "is presented with a registration page designed to capture the user's identity and preferred payment instrument details." EX1007, [0054]. As discussed in VII.A.3, instructions (e.g., through HTML or JavaScript) would be required to configure and render such registration form fields for inputting identity and payment instrument details.

- l) **[1.1.5.4] outputting instructions to the payer computing system, in response to the HTTP request for the generated URL, to encrypt the sensitive financial account information provided by the payer and transmit the encrypted financial account information to the secure server via the secure socket layer connection;**

94. In my opinion, Stringfellow/Kloster renders obvious Limitation 1.1.5.4]. As discussed in IX.A, in response to the merchant server's payment

authorization request message retrieving the sign-in URL, the user is presented with the URL where the user “login[s], or...register[s] with the trusted intermediary system” communications are transmitted using “HTTP over Secure Socket Layer protocol (HTTPS).” EX1007, [0045], [0074], [0075]. As discussed in **Error! Reference source not found.**, when transmitting data using “HTTP over Secure Socket Layer protocol (HTTPS),” a POSITA would understand that the URL presented to the payer computing system includes code instructing that system to make a secure “HTTPS” request specifying the URL as part of the “initial handshake.” Upon receipt of the HTTPS request, the trusted intermediary system sends an HTTPS response that contains additional instructions for the payer computer system, including chosen encryption method (“cipher_suite”) and “random” value that will be used in combination with the client random value to generate encryption keys. EX1012, 27/67. Following receipt of these encryption instructions and establishment of the secure SSL connection, a POSITA would understand that subsequent HTTP messages transmitted from the payer computing system would be encrypted in accordance with the SSL protocol before sending them across the Internet to the trusted intermediary system. Accordingly, this would include encrypting financial account information that is subsequently transmitted when the user “is presented with a registration page designed to capture the user’s

identity and preferred payment instrument details.” EX1007, [0054]. *See also* Limitation [1.2].

m) [1.2] receiving the sensitive financial account information provided by the payer via the secure socket layer connection;

95. In my opinion, Stringfellow/Kloster renders obvious Limitation [1.2]. As discussed for Limitation [1.1.5.4], when the user submits “a registration page designed to capture the user's identity and preferred payment instrument details,” the payer computer system transmits the information in the form over the SSL [secure socket layer] connection to the trusted intermediary. EX1007, [0054] (emphasis added). Stringfellow explains that “[t]he trusted intermediary system [] holds data in a database DB1 corresponding to users (buyers) and online merchants that have registered with the intermediary 10, together with transaction data...[including] a set of payment details for the user in the form of a stored set of records conveniently referred to as a remote store” *Id.*, [0039]. These payment details include “a bank account, a payment card or other account such as any payment instrument that can be given a unique account reference.” *Id.*, [0052].

n) [1.3] storing the sensitive financial account information in a secure storage location and performing each software process required to maintain compliance with one or more information security standards;

96. In my opinion, Stringfellow/Kloster renders obvious Limitation [1.3]. Stringfellow explains that “[t]he trusted intermediary system [] holds data in a

database DB1 corresponding to users (buyers) and online merchants that have registered with the intermediary 10, together with transaction data...[including] a set of payment details for the user in the form of a stored set of records conveniently referred to as a remote store” EX1007, [0039].

97. Stringfellow discloses that its payment system “has the benefit of reducing the risk of fraud that may be incurred in relation to conventional arrangements of payment systems.” *Id.*, [0028], [0078]. Stringfellow also teaches implementing security processes that enable compliance with known security standards through “[a]uthentication of a user into the trusted intermediary system 10 for payment transactions...according to any one of the 3 known categories...[including] 1-factor authentication... 2-factor authentication... 3-factor authentication. *Id.*, [0057]-[0061]. A POSITA would understand that including multi-factor authentication methods that rely on username/passwords, security questions/tokens, or biometrics, as taught by Stringfellow, ensures that only authorized individuals can access sensitive information or systems and enables compliance with security standards, including the Payment Card Industry (PCI) Data Security Standard. EX1024, 46-47 (discussing assignment of unique IDs to users and implementing multi-factor authentication).

98. Further, a POSITA would understand that compliance with one or more information security standards was required based on known regulations relating

transactions involving sensitive information. Kloster explains that “collecting such data [e.g., credit card numbers] may subject the online merchant to costly security audits and regulations.” Collison also explains that when systems capture payment information from their customers “this approach requires that the companies implement systems and policies to ensure the secure handling, transmission, and storage of their customers' sensitive payment information” and “[t]hese systems and policies must also be in compliance with the credit card networks as defined by various standards such as PCI Security Standard.” EX1021 1:45-60. Similarly, ’621 Patent acknowledges in its “Background” that “Annual PCI compliance audits by an external Qualified Security Assessor (QSA) are required for organizations handling large volumes of transactions.” EX1001, 1:36-39.

o) [1.4] executing a token retrieval function, upon initiation by the merchant server via the API, by:

99. In my opinion, Stringfellow/Kloster renders obvious Limitation [1.4]. As discussed in IX.D, token retrieval may occur either “the inline frame may be redirected to a call back URL *specified by the merchant*,” here Stringfellow’s online merchant system, such that “the payment data token is passed as a parameter to the call back URL” or “could be retrieved” through “API calls *by the merchant*.” EX1008, [0015], [0023], [0024]; EX1007, [0072]-[0073], [0076]. A POSITA would understand that both an “inline frame redirect” involving a “call back URL *specified by the merchant*” and an “API call *by the merchant*” are API requests initiated by

Stringfellow's "online merchant system" (the "merchant server") for Stringfellow's "trusted intermediary system" (the "secure server") to execute the token retrieval function. As explained in VII.A.5, API calls are a broad category of function calls made between servers. A POSITA would understand that requests made via a provided call back URL are merely one form of an API call. For example, Klein discloses an API in which "a subscribe/notify mechanism is provided allowing a client to start a subscription and, by providing a call back URL in the request, the server will be able to send a notification whenever there is a new event." EX1025, [0005]. In Klein, the client obtains a call back URL for communicating with an API via a "channel establishment (e.g., using HTTP Get, Put, and/or Delete)" function. *Id.*, [0051]. "The Callback URL is then used when sending notifications." *Id.*, [0055]. Thus, the call back URL is used as a vehicle for the client and API to interact ("call") each other.

- p) **[1.4.1] providing a non-sensitive electronic data token representing the sensitive financial account information to the merchant server without providing the sensitive financial account information to the merchant server and without providing the non-sensitive electronic data token to the payer; and**

100. In my opinion, Stringfellow/Kloster renders obvious Limitation [1.4.1]. Kloster's "payment data token" "is a system generated identifier comprised of a string of characters that may be utilized as a key to refer to sensitive payment data." EX1008, [0013]. "This payment data token only has meaning within the payment

system and could not be used to process payments at physical merchant locations.”

Id. As discussed with respect to Limitation [1.4], in either the “call back URL” or “API call” embodiments, the trusted intermediary system is instructed to provide the payment data token to the online merchant system. EX1008, [0004], [0015], [0021], [0023], [0024], [0025]. Further, Kloster discloses the online merchant system subsequently using the token exclusively as part of direct communications with the trusted intermediary system. EX1008, [0024], [0026]. Because Kloster does not disclose providing the token to the payer nor any application of use of the token that involves the payer, a POSTA would understand that, consistent with the fact that Kloster does not disclose doing so, it was, at a minimum obvious to not provide the data token to the payer because doing so would serve no purpose and would jeopardize security by unnecessarily exposing a token that can be used to request payments. For example, a bad actor obtaining the token may attempt to combine with other user data to request payment. Additionally, to the extent Kloster’s disclosure of a passing a payment data token as a parameter to the call back URL that redirects an “iframe” provided by the online merchant system is interpreted as the token being sent from the trusted intermediary system to the payer computer system, such functionality is still sufficient to meet the claim language, which precludes providing the token to the payer (i.e., the user), not the separately-recited payer computer system.

- q) **[1.4.2] processing the payment transaction using the sensitive financial account information by generating and transmitting an electronic request requesting the payment amount from the financial account, obtaining the payment amount, and forwarding at least a portion of the payment amount to the payee.**

101. In my opinion, Stringfellow/Kloster renders obvious Limitation [1.4.2].

As discussed in IX.D, in the Stingfellow/Kloster combination the trusted intermediary system/payment system processing the payment transaction by “receiving the payment data token from the business entity with the request; retrieving the sensitive payment data regarding the access to the monies held for the end user from the electronic database; and transferring the at least a portion of the monies held for the end user to the business entity utilizing the retrieved sensitive payment data via a backend payment gateway.” EX1008, [0004], [0015], [0023]-[0024].

2. *Claim 2*

- a) **[2.1] The method of claim 1, further comprising: receiving a second payment transaction instruction from the merchant server, the second payment transaction instruction including a representation of the non-sensitive electronic data token and specifying a second payment amount;**

102. In my opinion, Stringfellow/Kloster renders obvious Limitation [2.1] for reasons discussed above in Limitations [1.4.1], [1.4.2], and as discussed in IX.B and IX.D. Specifically, Kloster teaches that the business entity (i.e., the merchant server) may “utilize the same payment data token” and send “a second request...to transfer at least a second portion of the monies held for the end user to the business

entity ... [such that] the payment data token may be received from the business entity with the second request ... [and] the sensitive payment data regarding the access to the monies held for the end user may be retrieved from the electronic database.” EX1008, [0026].

- b) [2.2] processing the second payment by obtaining the second payment amount using the sensitive financial account information represented by the non-sensitive electronic data token, and forwarding at least a portion of the second payment amount to the payee.**

103. In my opinion, Stringfellow/Kloster renders obvious Limitation [2.2] for reasons discussed above in Limitation [2.1]. Kloster teaches processing the second payment amount based on the “payment data token...received from the business entity with the second request.” EX1008, [0026]. Specifically, “the sensitive payment data regarding the access to the monies held for the end user may be retrieved from the electronic database ... [and] a second portion of the monies held for the end user may be transferred to the business entity utilizing the retrieved sensitive payment data via the backend payment gateway.” EX1008, [0026].

3. Claim 3

- a) [3.1] The method of claim 2, wherein the representation of the non-sensitive electronic data token is the non-sensitive electronic data token.**

104. In my opinion, Stringfellow/Kloster renders obvious Limitation [3.1] for reasons discussed above in Limitation [2.1].

4. *Claim 4*

- a) **[4.1] The method of claim 1, wherein the non-sensitive electronic data token constitutes a symbolic representation of the sensitive financial account information.**

105. In my opinion, Stringfellow/Kloster renders obvious Limitation [4.1] for reasons discussed above in Limitation [1.4.1].

5. *Claim 5*

- a) **[5.1] The method of claim 1, wherein the at least one data element associated with the payer is a customer identification number generated by the merchant server to identify the payer.**

106. In my opinion, Stringfellow/Kloster renders obvious Limitation [5.1] for reasons discussed above in Limitation [1.1.3]. Stringfellow discloses that “[e]ach user has a profile, which comprises demographic and identification data for the user....” EX1007, [0052] (emphasis added). Further, Kloster discloses that “[a] unique customer identifier created by the merchant could also be stored [by the payment server 102] for the customer.” EX1008, [0015]. Such a customer identifier created by the merchant must necessarily be transmitted by the merchant system to the payment server 102. Based on these teachings, a POSITA would understand it would have been obvious for the merchant server to send the payment amount and customer identifier and, once received, for the payment server to use the customer identifier to identify the payer, as disclosed by Kloster.

6. *Claim 6*

- a) **[6.1] The method of claim 1, wherein the secure socket layer connection is synchronous or asynchronous with a process, performed by the merchant server, to receive enrollment data provided by the payer.**

107. In my opinion, Stringfellow/Kloster renders obvious Limitation [6.1]. First, as discussed for Limitation [1.1.5.2], Stringfellow renders obvious providing a secure socket layer connection between the secure server and the payer computing system.

108. Second, Stringfellow discloses that a “user completes their shopping experience with the online merchant system, initiates checkout using the online merchant system, and proceeds to the virtual checkout, according to conventional methods available through commonly available shopping cart and check-out software packages such as are known to the skilled person.” EX1007, [0045]. *see also id.*, [0028] (“the relationship for such transactions is between the buyer and the online merchant”). A POSITA would understand that that “conventional” “virtual checkout” includes the merchant server collecting and receiving enrollment data from the payer (e.g., information about the payer such as name and shipping details). Consistent with this understanding, Stringfellow describes an embodiment in which “the checkout functionality could be provided by the online merchant system,” including collecting and “storing shipping details.” EX1007, [0081].

109. As discussed in IX.A, the virtual checkout process precedes the online merchant system requesting “the URL corresponding to a sign-in page of the trusted intermediary system” that subsequently allows for the “creation of a secure session” as discussed in Limitation [1.1.5.2]. EX1007, [0075]. Thus, in the terminology of the ’621 Patent, those two processes are “asynchronous.” EX1001, 9:36-40, 10:27-30.

7. Claim 7

- a) **[7.1] The method of claim 1, wherein the sensitive financial account information includes at least one of a credit card number or bank account number.**

110. In my opinion, Stringfellow/Kloster renders obvious Limitation [7.1]. See Limitation [1.2].

8. Claim 8

- a) **[8.Pre] A secure server that processes a payment transaction from a payer to a payee, the secure server comprising:**

111. In my opinion, Stringfellow/Kloster renders obvious Limitation [8.Pre] for reasons discussed above in Limitation [1.Pre] and below in [8.1].

- b) **[8.1] one or more processors; and**

112. In my opinion, Stringfellow/Kloster renders obvious Limitation [8.1]. Stringfellow discloses that the trusted intermediary system includes at least one “data processing node...a server computer,” which a POSITA would understand includes at least one processor. EX1007, [0082]. Stringfellow explains that “the term

‘system’, when applied to entities such as ... the trusted intermediary system ... should be understood to mean a data processing function, provided at one or more physical sites, connected to other data processing functions via data communications links...[and] [e]ach function may be provided by a single data processing node, for example a server computer, or a set of data processing nodes providing fail-over backup to each other, such as a cluster of server computers.” EX1007, [0082]. A POSITA would understand that a “data processing node...a server computer,” necessarily includes a processor responsible for executing instruction necessary to handle tasks like managing data, executing applications, and processing requests.

c) [8.2] memory storing instructions that, when executed by the one or more processors, cause the secure server to:

113. In my opinion, Stringfellow/Kloster renders obvious Limitation [8.2]. A POSITA would understand that Stringfellow requires memory storing instruction to “manage[] and provide[] access to the common business logic of the platform, and [provide] a web server & J2EE servlet engine 403, which acts as the entry point for external HTTP requests to the trusted intermediary system 10 from online merchants and from users' browsers.” EX1007, [0072]. Specifically, Stringfellow explains “the trusted intermediary system 10 is preferably embodied as a web application server, for example as a J2EE compliant application server 401 which manages and provides access to the common business logic of the platform, and a web server & J2EE servlet engine 403, which acts as the entry point for external

HTTP requests to the trusted intermediary system 10 from online merchants and from users' browsers.” EX1007, [0072]. “The web server and servlet engine 403 comprises presentation components, which expose web services-based payment APIs or API wrappers to online merchant systems...[and] comprises presentation processing components 404 which are configured to generate and manage the interface to the user, e.g. when the user selects a payment method in the manner described above.” EX1007, [0072]. Further, the “J2EE Application Server 401 manages all the business logic for the web platform and applications” where “[t]he business logic comprises functional software components 411 a . . . 411 e.” EX1007, [0073], [0074]. Based on this disclosure, a POSITA would understand that the trusted intermediary system necessarily includes memory to store instructions “configured to generate and manage” the API interface for the users and merchant system, as well as for executing the “logic of the platform.”

- d) [8.2.1] provide an application programming interface (API) to a merchant server providing a webpage to a payer computing system used by the payer, the API configured to:**

114. In my opinion, Stringfellow/Kloster renders obvious Limitation [8.2.1].

See Limitation [1.1].

- e) [8.2.1.1] provide financial account registration and token retrieval functions that can be executed to process the payment transaction;**

115. In my opinion, Stringfellow/Kloster renders obvious Limitation [8.2.1.1]. See Limitation [1.1.1].

- f) **[8.2.1.2] provide access to the financial account registration and token retrieval functions to the merchant server;**

116. In my opinion, Stringfellow/Kloster renders obvious Limitation [8.2.1.2]. See Limitation [1.1.2].

- g) **[8.2.1.3] receive, from the merchant server via the API, at least one data element associated with the payer and a payment amount from the payer to the payee;**

117. In my opinion, Stringfellow/Kloster renders obvious Limitation [8.2.1.3]. See Limitation [1.1.3].

- h) **[8.2.1.4] authenticate the payee;**

118. In my opinion, Stringfellow/Kloster renders obvious Limitation [8.2.1.4]. See Limitation [1.1.4].

- i) **[8.2.1.5] execute the financial account registration function, upon initiation by the merchant server, by:**

119. In my opinion, Stringfellow/Kloster renders obvious Limitation [8.2.1.5]. See Limitation [1.1.5].

- j) **[8.2.1.5.1] generating a uniform resource locator (URL) to establish a secure socket layer connection via the internet between the secure server and the payer computing system, the URL comprising either:**

120. In my opinion, Stringfellow/Kloster renders obvious Limitation [8.2.1.5.1]. See Limitation [1.1.5.1].

- k) **[8.2.1.5.1a] [the URL comprising either] a dynamic URL generated by the secure server for the payer and the payee; or a static URL and a hypertext transport protocol (HTTP) parameter used by the secure server to identify the payer and the payee;**

121. In my opinion, Stringfellow/Kloster renders obvious Limitation [8.2.1.5.1a]. See Limitation [1.1.5.1a].

- l) **[8.2.1.5.2] establishing the secure socket layer connection, in response to an HTTP request received from the merchant server for the generated URL, between the secure server and the payer computing system within a window or frame that is displayed within the webpage provided by the merchant server;**

122. In my opinion, Stringfellow/Kloster renders obvious Limitation [8.2.1.5.2]. See Limitation [1.1.5.2].

- m) **[8.2.1.5.3] outputting instructions to the payer computing system, in response to the HTTP request for the generated URL, to render a financial account registration request form, within the window or frame that is displayed within the webpage provided by the merchant server, that provides functionality for the payer to provide sensitive financial account information associated with a financial account; and**

123. In my opinion, Stringfellow/Kloster renders obvious Limitation [8.2.1.5.3]. See Limitation [1.1.5.3].

- n) **[8.2.1.5.4] outputting instructions to the payer computing system, in response to the HTTP request for the generated URL, to encrypt the sensitive financial account information provided by the payer and transmit the encrypted financial account information to the secure server via the secure socket layer connection;**

124. In my opinion, Stringfellow/Kloster renders obvious Limitation [8.2.1.5.4]. See Limitation [1.1.5.4].

- o) [8.2.1.6] receive the sensitive financial account information provided by the payer via the secure socket layer connection;**

125. In my opinion, Stringfellow/Kloster renders obvious Limitation [8.2.1.6]. See Limitation [1.2].

- p) [8.2.1.7] store the sensitive financial account information in a secure storage location and perform each software process required to maintain compliance with one or more information security standards;**

126. In my opinion, Stringfellow/Kloster renders obvious Limitation [8.2.1.7]. See Limitation [1.3].

- q) [8.2.1.8] execute a token retrieval function, upon initiation by the merchant server by the API, by:**

127. In my opinion, Stringfellow/Kloster renders obvious Limitation [8.2.1.8]. See Limitation [1.4].

- r) [8.2.1.8.1] providing a non- sensitive electronic data token representing the sensitive financial account information to the merchant server without providing the sensitive financial account information to the merchant server and without providing the non-sensitive electronic data token to the payer; and**

128. In my opinion, Stringfellow/Kloster renders obvious Limitation [8.2.1.8.1]. See Limitation [1.4.1].

- s) **[8.2.1.8.2] processing the payment transaction using sensitive the financial account information by generating and transmitting an electronic request requesting the payment amount from the financial account, obtaining the payment amount, and forwarding at least a portion of the payment amount to the payee.**

129. In my opinion, Stringfellow/Kloster renders obvious Limitation [8.2.1.8.2]. See Limitation [1.4.2].

9. Claim 9

- a) **[9.1] The secure server of claim 8, wherein the instructions further cause the secure computing system to: receive a second payment transaction instruction from the merchant server, wherein the second payment transaction instruction includes a representation of the non- sensitive electronic data token and specifies a second payment amount;**

130. In my opinion, Stringfellow/Kloster renders obvious Limitation [9.1]. See Limitation [2.1]. For the server of Claim 8, *see supra*, Limitation [8.1] and, for instructions causing the act, *see supra*, Limitation [8.2].

- b) **[9.2] process the second payment by obtaining the second payment amount using the sensitive financial account information represented by the non- sensitive electronic data token; and forward at least a portion of the second payment amount to the payee.**

131. In my opinion, Stringfellow/Kloster renders obvious “process the second payment by obtaining the second payment amount using the sensitive financial account information represented by the non- sensitive electronic data token.” See Limitation [2.2].

132. In my opinion, Stringfellow/Kloster renders obvious “forward at least a portion of the second payment amount to the payee.” See Limitation [2.2].

10. Claim 10

- a) **[10.1] The secure server of claim 9, wherein the representation of the non-sensitive electronic data token is the non-sensitive electronic data token.**

133. In my opinion, Stringfellow/Kloster renders obvious Limitation [10.1]. See Limitation [3.1]. For the server of Claim 9, *see supra*, Limitation [8.1].

11. Claim 11

- a) **[11.1] The secure server of claim 8, wherein the non-sensitive electronic data token constitutes a symbolic representation of the sensitive financial account information.**

134. In my opinion, Stringfellow/Kloster renders obvious Limitation [11.1]. See Limitation [4.1]. For the server of Claim 8, *see supra*, Limitation [8.1].

12. Claim 12

- a) **[12.1] The secure server of claim 8, wherein the at least one data element associated with the payer is a customer identification number generated by the merchant server to identify the payer.**

135. In my opinion, Stringfellow/Kloster renders obvious Limitation [12.1]. See Limitation [5.1]. For the server of Claim 8, *see supra*, Limitation [8.1].

13. Claim 13

- a) **[13.1] The secure server of claim 8, wherein the secure socket layer connection is synchronous or asynchronous with a process, performed by the merchant server, to receive enrollment data from the payer.**

136. In my opinion, Stringfellow/Kloster renders obvious Limitation [13.1].

See Limitation [6.1]. For the server of Claim 8, *see supra*, Limitation [8.1].

14. Claim 14

- a) **[14.1] The secure server of claim 8, wherein the sensitive financial account information includes at least one of a credit card number or bank account number.**

137. In my opinion, Stringfellow/Kloster renders obvious Limitation [14.1].

See Limitation [7.1]. For the server of Claim 8, *see supra*, Limitation [8.1].

15. Claim 15

- a) **[15.Pre] A method of enrolling a payer by a merchant server operated by or for the benefit of a payee, the method comprising:**

138. In my opinion, Stringfellow/Kloster renders obvious Limitation [15.Pre]. See Limitations [1.Pre], [5.1], and [6.1].

- b) **[15.1] providing a webpage to a payer computing system used by a payer;**

139. In my opinion, Stringfellow/Kloster renders obvious Limitation [15.1].

See Limitation [1.1].

- c) **[15.2] receiving enrollment data identifying the payer;**

140. In my opinion, Stringfellow/Kloster renders obvious Limitation [15.2].

See Limitations [1.1.3], [5.1] and [6.1].

- d) **[15.3] providing, to a secure server via an application programming interface (API) provided by the secure server, at least one data element associated with the payer and a payment amount from the payer to the payee;**

141. In my opinion, Stringfellow/Kloster renders obvious Limitation [15.3].

See Limitation [1.1.3].

- e) **[15.4] displaying a window or frame within the webpage provided to the payer computing system;**

142. In my opinion, Stringfellow/Kloster renders obvious Limitation [15.4].

See Limitation [1.1.5.2].

- f) **[15.5] providing authentication credentials to the API to establish an authenticated session, the authentication credentials identifying the payee and authenticating the payee;**

143. In my opinion, Stringfellow/Kloster renders obvious Limitation [15.5].

See Limitations [1.1.3] and [1.1.4].

- g) **[15.6] initiating a financial account registration function of the API during the authenticated session, the financial account registration function;**

144. In my opinion, Stringfellow/Kloster renders obvious Limitation [15.6].

See Limitation [1.1.5].

- h) **[15.6.1] establishing a secure socket layer connection between the secure server and the payer computing system within the window or frame;**

145. In my opinion, Stringfellow/Kloster renders obvious Limitation [15.6.1]. See Limitation [1.1.5.2].

- i) **[15.6.2] outputting instructions to the payer computing system to render a financial account registration request form, within the window or frame, that provides functionality for the payer to provide sensitive financial account information associated with a financial account; and**

146. In my opinion, Stringfellow/Kloster renders obvious Limitation [15.6.2]. See Limitation [1.1.5.3].

- j) **[15.6.3] outputting instructions to the payer computing system to encrypt the sensitive financial account information provided by the payer and transmit the encrypted financial account information to the secure server, via the secure socket layer connection, without providing the sensitive financial account information to the merchant server;**

147. In my opinion, Stringfellow/Kloster renders obvious Limitation [15.6.3]. See Limitation [1.1.5.4].

- k) **[15.7] initiating a token retrieval function of the API during the authenticated session, the token retrieval function outputting a non- sensitive electronic data token representing the sensitive financial account information to the merchant server;**

148. In my opinion, Stringfellow/Kloster renders obvious Limitation [15.7]. See Limitations [1.4] and [1.4.1].

- l) **[15.8] receiving the non- sensitive electronic data token from the secure server, wherein the merchant server does not receive the sensitive financial account information represented by the non-sensitive electronic data token and the secure server does not provide the non-sensitive electronic data token to the payer;**

149. In my opinion, Stringfellow/Kloster renders obvious Limitation [15.8]. See Limitation [1.4.1].

m) [15.9] storing the non- sensitive electronic data token in association with the enrollment data identifying the payer; and

150. In my opinion, Stringfellow/Kloster renders obvious Limitation [15.9]. See Limitations [1.4.1], [2.1], and [6.1]. Stringfellow teaches that enrollment data would be obtained when a “user completes their shopping experience with online merchant C's online merchant system, initiates checkout using the online merchant system, and proceeds to the virtual checkout, according to conventional methods available through commonly available shopping cart and check-out software packages such as are known to the skilled person.” EX1007, [0045]. Further, Kloster teaches that “if the online merchant would need to collect a payment again utilizing the same payment data (e.g., to renew a monthly subscription), the merchant may do so utilizing the same payment data token.” EX1008, [0026]. Accordingly, a POSITA would recognize that when a merchant server initiates repeated transactions (e.g., monthly subscriptions), it requires storing a data token that is associated with user provided information obtained during the initial checkout process. The merchant server must have some way of identifying the payers that are charged in each subsequent transaction. Before tokens, this was sometimes done by storing the payer's sensitive payment information directly. But, as explained, this subjected merchants to additional compliance requirements. As an alternative to storing sensitive data for subsequent transactions, merchants can instead store the non-sensitive electronic data token, as taught by Kloster.

- n) **[15.10] processing a payment from the payer by generating a payment transaction instruction, using the token retrieval function of the API, that includes a representation of the non-sensitive electronic data token and instructs the secure server to generate and transmit an electronic request requesting the payment amount from the financial account, obtain the payment amount, and forward at least a portion of the payment amount.**

151. In my opinion, Stringfellow/Kloster renders obvious Limitation [15.10]. See Limitations [1.4], [1.4.1], and [1.4.2]. Specifically, Kloster teaches generating a payment transaction instruction through “a request [that] is received from the business entity to transfer at least a portion of the monies held for the end user to the business entity” that uses the retrieved token where “the payment data token is received from the business entity with the request.” EX1008, [0024], [0012]. Further, the request instructs the payment system server to retrieve “[t]he sensitive payment data regarding the access to the monies held for the end user” and transfer “a portion of the monies held for the end user...to the business entity utilizing the retrieved sensitive payment data.” EX1008, [0024], [0012].

16. Claim 16

- a) **[16.1] The method of claim 15, wherein the representation of the non-sensitive electronic data token is the non-sensitive electronic data token.**

152. In my opinion, Stringfellow/Kloster renders obvious Limitation [16.1]. See Limitation [3.1].

17. Claim 17

- a) **[17.1] The method of claim 15, wherein the non-sensitive electronic data token constitutes a symbolic representation of the sensitive financial account information.**

153. In my opinion, Stringfellow/Kloster renders obvious Limitation [17.1].

See Limitation [4.1]. For the method of Claim 15, *see supra*, Limitation [15.Pre].

18. Claim 18

- a) **[18.1] The method of claim 15, wherein the at least one data element associated with the payer is a customer identifier generated by the merchant server to identify the payer.**

154. In my opinion, Stringfellow/Kloster renders obvious Limitation [18.1].

See Limitation [5.1].

19. Claim 19

- a) **[19.1] The method of claim 15, wherein the secure socket layer connection provides access to the secure server using a static or dynamic uniform resource locator (URL).**

155. In my opinion, Stringfellow/Kloster renders obvious Limitation [19.1].

See Limitation [1.1.5.1a]. For the method of Claim 15, *see supra*, Limitation [15.Pre].

20. Claim 20

- a) **[20.1] The method of claim 15, wherein the secure socket layer connection is synchronous or asynchronous with a process, performed by the merchant server, to receive enrollment data from the payer.**

156. In my opinion, Stringfellow/Kloster renders obvious Limitation [20.1].

See Limitation [6.1]. For the method of Claim 15, *see supra*, Limitation [15.Pre].

21. Claim 21

- a) **[21.1] The method of claim 15, wherein the enrollment data identifying the payer is received from the payer over a network.**

157. In my opinion, Stringfellow/Kloster renders obvious Limitation [21.1].

See Limitation [6.1]. For the method of Claim 15, *see supra*, Limitation [15.Pre].

22. Claim 22

- a) **[22.1] The method of claim 15, wherein the sensitive financial account information includes at least one of a credit card number or bank account number.**

158. In my opinion, Stringfellow/Kloster renders obvious Limitation [22.1].

See Limitation [7.1]. For the method of Claim 15, *see supra*, Limitation [15.Pre].

23. Claim 23

- a) **[23.Pre] A merchant server, operated by or for the benefit of a payee, comprising:**

159. In my opinion, Stringfellow/Kloster renders obvious Limitation [23.Pre]. See Limitation [15.Pre].

- b) **[23.1] one or more computer processors; and**

160. In my opinion, Stringfellow/Kloster renders obvious Limitation [23.1].
See Limitation [8.1].

- c) **[23.2] non-transitory computer readable storage media storing instructions that, when executed by the one or more computer processors, cause the merchant server to:**

161. In my opinion, Stringfellow/Kloster renders obvious Limitation [23.2].
See Limitation [8.2].

- d) **[23.2.1] provide a webpage to a payer computing system used by a payer;**

162. In my opinion, Stringfellow/Kloster renders obvious Limitation [23.2.1]. See Limitation [15.1].

- e) **[23.2.2] receive enrollment data identifying the payer;**

163. In my opinion, Stringfellow/Kloster renders obvious Limitation [23.2.2]. See Limitation [15.2].

- f) **[23.2.3] provide, to a secure server via an application programming interface (API) provided by the secure server, at least one data element associated with the payer and a payment amount from the payer to the payee;**

164. In my opinion, Stringfellow/Kloster renders obvious Limitation [23.2.3]. See Limitation [15.3].

- g) **[23.2.4] display a window or frame within the webpage provided to the payer computing system;**

165. In my opinion, Stringfellow/Kloster renders obvious Limitation [23.2.4]. See Limitation [15.4].

- h) **[23.2.5] provide authentication credentials to the API to establish an authenticated session, the authentication credentials identifying the payee and authenticating the payee;**

166. In my opinion, Stringfellow/Kloster renders obvious Limitation [23.2.5]. See Limitation [15.5].

- i) **[23.2.6] initiate a financial account registration function of the API during the authenticated session, the financial account registration function:**

167. In my opinion, Stringfellow/Kloster renders obvious Limitation [23.2.6]. See Limitation [15.6].

- j) **[23.2.6.1] establishing a secure socket layer connection between the secure server and the payer computing system within the window or frame;**

168. In my opinion, Stringfellow/Kloster renders obvious Limitation [23.2.6.1]. See Limitation [15.6.1].

- k) **[23.2.6.2] outputting instructions to the payer computing system to render a financial account registration request form, within the window or frame, that provides functionality for the payer to provide sensitive financial account information associated with a financial account; and**

169. In my opinion, Stringfellow/Kloster renders obvious Limitation [23.2.6.2]. See Limitation [15.6.2].

- l) **[23.2.6.3] outputting instructions to the payer computing system to encrypt the sensitive financial account information provided by the payer and transmit the encrypted financial account information to the secure server via the secure socket layer connection;**

170. In my opinion, Stringfellow/Kloster renders obvious Limitation [23.2.6.3]. See Limitation [15.6.3].

- m) **[23.2.7] initiate a token retrieval function of the API during the authenticated session, the token retrieval function outputting a non-sensitive electronic data token representing the sensitive financial account information to the merchant server;**

171. In my opinion, Stringfellow/Kloster renders obvious Limitation [23.2.7]. See Limitation [15.7].

- n) **[23.2.8] receive the non- sensitive electronic data token from the secure server, wherein the merchant server does not receive the sensitive financial account information represented by the non-sensitive electronic data token and the secure server does not provide the non-sensitive electronic data token to the payer;**

172. In my opinion, Stringfellow/Kloster renders obvious Limitation [23.2.8]. See Limitation [15.8].

- o) **[23.2.9] store the non-sensitive electronic data token in association with the enrollment data identifying the payer; and**

173. In my opinion, Stringfellow/Kloster renders obvious Limitation [23.2.9]. See Limitation [15.9].

- p) **[23.2.10] process a payment from the payer by generating a payment transaction instruction, using the token retrieval function of the API, that includes a representation of the non-sensitive electronic data token and instructs the secure server to generate and transmit an electronic request requesting the payment amount from the financial account, obtain the payment amount, and forward at least a portion of the payment amount.**

174. In my opinion, Stringfellow/Kloster renders obvious Limitation [23.2.10]. See Limitation [15.10].

24. Claim 24

- a) **[24.1] The merchant server of claim 23, wherein the non-sensitive electronic data token constitutes a symbolic representation of the sensitive financial account information.**

175. In my opinion, Stringfellow/Kloster renders obvious Limitation [24.1].

See Limitation [4.1]. For the server of Claim 23, *see supra*, IX.E.23.

25. Claim 25

- a) **[25.Pre] A method of processing a payment transaction from a payer to a payee, the method being performed by one or more secure servers, the method comprising:**

176. In my opinion, Stringfellow/Kloster renders obvious Limitation [25.Pre]. See Limitation [1.Pre].

- b) **[25.1] providing, by the one or more secure servers to a merchant server providing a webpage to a payer computing system used by the payer, an application programming interface (API) that:**

177. In my opinion, Stringfellow/Kloster renders obvious Limitation [25.1].
See Limitation [1.1].

- c) **[25.1.1] provides financial account registration and token retrieval functions that can be executed to process the payment transaction;**

178. In my opinion, Stringfellow/Kloster renders obvious Limitation [25.1.1]. See Limitation [1.1.1].

- d) **[25.1.2] provides access to the financial account registration and token retrieval functions to the merchant server;**

179. In my opinion, Stringfellow/Kloster renders obvious Limitation [25.1.2]. See Limitation [1.1.2].

- e) **[25.1.3] receives, from the merchant server via the API, at least one data element associated with the payer and a payment amount from the payer to the payee;**

180. In my opinion, Stringfellow/Kloster renders obvious Limitation [25.1.3]. See Limitation [1.1.3].

- f) **[25.1.4] authenticates the payee; and**

181. In my opinion, Stringfellow/Kloster renders obvious Limitation [25.1.4]. See Limitation [1.1.4].

- g) **[25.1.5] executes the financial account registration function, upon initiation by the merchant server, by:**

182. In my opinion, Stringfellow/Kloster renders obvious Limitation [25.1.5]. See Limitation [1.1.5].

- h) **[25.1.5.1] generating a uniform resource locator (URL), for establishing a secure socket layer connection via the internet between the secure server and the payer computing system, the URL comprising either:**

183. In my opinion, Stringfellow/Kloster renders obvious Limitation [25.1.5.1]. See Limitation [1.1.5.1].

- i) **[23.1.5.1a] [the URL comprising either] a dynamic URL generated by the secure server for the payer and the payee; or a static URL and a hypertext transport protocol (HTTP) parameter used by the secure server to identify the payer and the payee;**

184. In my opinion, Stringfellow/Kloster renders obvious Limitation [23.1.5.1a]. See Limitation [1.1.5.1a].

- j) **[25.1.5.2] establishing the secure socket layer connection, in response to an HTTP request received for the generated URL, between the secure server and the payer computing system;**

185. In my opinion, Stringfellow/Kloster renders obvious Limitation [25.1.5.2]. See Limitation [1.1.5.2].

- k) **[25.1.5.3] outputting instructions to the payer computing system, in response to the HTTP request for the generated URL, to render a financial account registration request form that provides functionality for the payer to provide sensitive financial account information associated with a financial account; and**

186. In my opinion, Stringfellow/Kloster renders obvious Limitation [25.1.5.3]. See Limitation [1.1.5.3].

- l) **[25.1.5.4] outputting instructions to the payer computing system, in response to the HTTP request for the generated URL, to encrypt the financial account information provided by the payer and transmit the encrypted financial account information to the secure server, via the secure socket layer connection;**

187. In my opinion, Stringfellow/Kloster renders obvious Limitation [25.1.5.4]. See Limitation [1.1.5.4].

- m) **[25.2] receiving the sensitive financial account information provided by the payer via the secure socket layer connection;**

188. In my opinion, Stringfellow/Kloster renders obvious Limitation [25.2].
See Limitation [1.2].

- n) **[25.3] storing the sensitive financial account information in a secure storage location and performing each software process required to maintain compliance with one or more information security standards;**

189. In my opinion, Stringfellow/Kloster renders obvious Limitation [25.3].

See Limitation [1.3].

- o) **[25.4] executing a token retrieval function, upon initiation by the merchant server via the API, by:**

190. In my opinion, Stringfellow/Kloster renders obvious Limitation [25.4].

See Limitation [1.4].

- p) **[25.4.1] providing a non- sensitive electronic data token representing the sensitive financial account information to the merchant server without providing the sensitive financial account information to the merchant server and without providing the non-sensitive electronic data token to the payer; and**

191. In my opinion, Stringfellow/Kloster renders obvious Limitation [25.4.1]. See Limitation [1.4.1].

- q) **[25.4.2] processing the payment transaction using the sensitive financial account information by generating and transmitting an electronic request requesting the payment amount from the financial account, obtaining the payment amount, and forwarding at least a portion of the payment amount to the payee.**

192. In my opinion, Stringfellow/Kloster renders obvious Limitation [25.4.2]. See Limitation [1.4.2].

26. Claim 26

- a) **[26.1] The method of claim 25, wherein the URL is a redirect URL that redirects the payer computing system to the secure server.**

193. In my opinion, Stringfellow/Kloster renders obvious Limitation [26.1].

See Limitation [1.1.5.1]. For the method of Claim 25, see supra, Limitation [25.Pre].

27. Claim 27

- a) **[27.1] The method of claim 25, wherein the non-sensitive electronic data token constitutes a symbolic representation of the sensitive financial account information.**

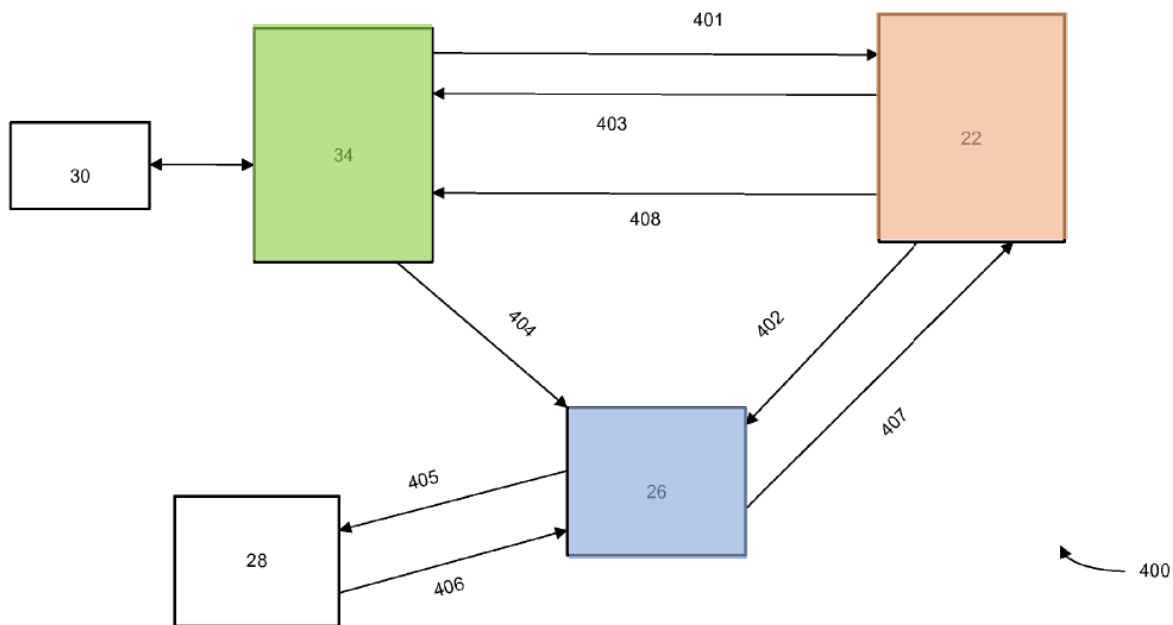
194. In my opinion, Stringfellow/Kloster renders obvious Limitation [27.1].

See Limitation [4.1]. For the method of Claim 25, see supra, Limitation [25.Pre].

X. GROUND 2: STRINGFELLOW, KLOSTER, AND CARLSON

A. Carlson

195. Carlson, like Stringfellow and Kloster, discloses a method “for handling and/or authorizing payment requests” by presenting a web page on “access device 34” (green, the claimed “payer computer system”), where a first server, “merchant 22” (red, meets the claimed “merchant server”), provides content for an online shopping experience and “an IFrame,” “the new payment frame,” “is sent from payment processing network 26 (blue, meets the claimed “one or more secure servers”). EX1018, Abstract, [0056], [0063], FIG. 4.



Id., Fig. 4 (annotated).

196. Carlson likewise discloses “payment processing network 26” communication with each of “access device 34” and “merchant server 22” over the course of a transaction. *Id.* To link these disparate messages to the same transaction, Carlson discloses that “merchant server 22 sends the transaction information to the payment processing network 26,” which “includes a merchant identification (ID), an amount of the transaction, and **a correlation ID, which ties together messages associated with the same transaction.**” *Id.*, [0080], [0042]. Subsequently, “at least some of the transaction information,” which may include “a merchant ID” and a “correlation ID,” is sent to the payment processing network 26” by “access device 34” when a link “is selected.” *Id.*, [0070]. Carlson also discloses code for sending the merchant ID and correlation ID from access device 34 to “a destination address

linked to the payment processing network 26” in a query string of a URL requesting code to populate an iFrame: “<iframe src= ‘http://www.ivisa.com/zifpvisa/php?Merchid=2000Scorrid=123 ...>.” *Id.*, [0100].

B. Motivation to Combine

197. As discussed in X.A.1, a POSITA would understand that a communication from the payer computer system to the trusted intermediary must somehow be tied to the transaction previously started by the online merchant system. A POSITA would be motivated to include such information in a query string of the sign-in URL based on Colson’s explicit explanation that providing the same transaction information in multiple messages to a secure server ***which ties together messages associated with the same transaction.***” EX1018, [0080], [0042]

198. A POSITA would have understood that storing data from past transactions (to the extent allowed by compliance standards) would help improve the customer experience by streamlining the checkout process. Additionally, a POSITA would have understood that using the well known technique of passing messages using the query string of a URL would be a useful tool in this context. Further, improved coordination between the online merchant and the trusted intermediary enhances the continuity and security of the transaction process, helping the payment server confirm that the user who initiates the payment is associated with the online merchant’s transaction session with the trusted intermediary.

C. Modification of Stringfellow in view of Kloster and Carlson

199. Applying Carlson's teachings to the Stringfellow/Kloster combination results in the same system as already described in IX in which the query string of the sign-in URL includes the same two transaction identifiers to the trusted intermediary in the original request message: "the online merchant account identifier and an identifier for the order." EX1007, [0045].

D. Element-By-Element Analysis

200. For the reasons discussed in X.A, X.B, and X.C, the Stringfellow/Kloster combination, in view of Carlson, renders obvious Claims 1-27. Additionally, because the modification of Stringfellow in view of Kloster and Carlson results in the same system as described for Ground 1, the same analysis applies.

XI. CONCLUSION

201. All statements made herein of my own knowledge are true, all statements made herein on information and belief are believed to be true, and these statements were made with the knowledge that willful false statement and the like are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001.

Respectfully submitted,

Dated: November 6, 2024



Kevin Almeroth, Ph.D.