

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

NUVEI TECHNOLOGIES, INC.

Petitioner,

v.

AUTOSCRIBE CORPORATION,

Patent Owners.

Case: IPR2025-00089
U.S. Patent No. 11,620,621

PETITION FOR *INTER PARTES* REVIEW UNDER 35 U.S.C. § 312

TABLE OF CONTENTS

TABLE OF CONTENTS.....	ii
TABLE OF EXHIBITS.....	iv
CLAIM LISTING	vi
I. INTRODUCTION	1
II. MANDATORY NOTICES UNDER 37 C.F.R. §42.8.....	4
A. REAL PARTY IN INTEREST (37 C.F.R. §42.8(B)(1)).....	4
B. RELATED MATTERS (37 C.F.R. §42.8(B)(2)).....	4
1. <i>Litigation</i>	4
2. <i>Administrative Proceedings</i>	5
C. DESIGNATION OF COUNSEL (37 C.F.R. §42.8(B)(3)).....	5
D. SERVICE INFORMATION (37 C.F.R. §42.8(B)(4)).....	5
III. GROUNDS FOR STANDING.....	6
IV. THE GROUNDS IN THIS PETITION ARE NOT CUMULATIVE.....	6
A. ORIGINAL PROSECUTION	6
B. IPR2024-01159	7
V. STATEMENT OF PRECISE RELIEF REQUESTED FOR EACH CLAIM CHALLENGED	8
A. CLAIMS FOR WHICH REVIEW IS REQUESTED (37 C.F.R. § 42.104(B)(1)).....	8
B. STATUTORY GROUNDS OF CHALLENGE (§42.104(B)(2)).....	8
VI. LEVEL OF ORDINARY SKILL IN THE ART	9
VII. FIELD OF TECHNOLOGY	10
A. HTTP COMMUNICATIONS.....	10
1. <i>HTTP Communications</i>	10
2. <i>Conveying State/Context Information In Stateless HTTP</i>	10
3. <i>HTML and iFrames</i>	11
4. <i>HTTPS / Secure Socket Layer (“SSL”) Connections</i>	11
5. <i>Application Programming Interface (API)</i>	12

B.	ONLINE SHOPPING AND PAYMENTS	13
1.	<i>Online Shopping and Payment Processing</i>	13
2.	<i>Tokenization</i>	14
VIII.	THE '621 PATENT	15
A.	OVERVIEW	15
B.	PROSECUTION HISTORY	17
IX.	CLAIM CONSTRUCTION	18
X.	REASONS FOR THE RELIEF REQUESTED (§§42.22(a)(2), 42.104(b)(4)) ..	20
A.	GROUND 1: STRINGFELLOW AND KLOSTER RENDER OBVIOUS CLAIMS 1-27.....	20
1.	<i>Stringfellow</i>	20
2.	<i>Kloster</i>	28
3.	<i>Motivation to Combine</i>	30
4.	<i>Modification of Stringfellow in view of Kloster</i>	33
5.	<i>Element-by-Element Analysis</i>	36
B.	GROUND 2: STRINGFELLOW, KLOSTER, AND CARLSON RENDER OBVIOUS CLAIMS 1-27.....	70
1.	<i>Carlson</i>	70
2.	<i>Motivation to Combine</i>	72
3.	<i>Modification of Stringfellow in view of Kloster and Carlson</i> ...	72
4.	<i>Element-By-Element Analysis</i>	72
XI.	CONCLUSION.....	73
	CERTIFICATE OF SERVICE	75

TABLE OF EXHIBITS

Exhibit	Short Name	Description
EX1001	'621 Patent	U.S. Patent No. 11,620,621
EX1002	'621 File History	File History of U.S. Patent No. 11,620,621
EX1003	'185 File History	File History of U.S. Patent Application No. 15/408,185
EX1004	'545 File History	File History of U.S. Patent Application No. 13/679,545
EX1005	Almeroth Declaration	Declaration of Dr. Kevin Almeroth
EX1006	Almeroth CV	<i>Curriculum Vitae</i> of Dr. Keven Almeroth
EX1007	Stringfellow	U.S. Publication No. 2010/0174626 A1
EX1008	Kloster	U.S. Publication No. 2010/0094755 A1
EX1009	RFC 1738	Tim Berners-Lee et al., Request for Comments: 1738 (December 1994)
EX1010	RFC 2396	Tim Berners-Lee et al., Request for Comments: 2396 (August 1998)
EX1011	RFC 2616	Roy Fielding et al., Request for Comments: 2616 (June 1999)
EX1012	RFC 6101	Alan Freier et al., Request for Comments: 6101 (August 2011)
EX1013	Boyd	U.S. Pat. No. 8,126,778
EX1014	Raleigh	U.S. Pat. No. 10,248,996
EX1015	Petri	U.S. Pat. No. 9,785,988
EX1016	Nuvei Complaint	Doc. No. 1, Complaint, <i>Autoscribe Corp. v. Nuvei Corp., et al.</i> , No. 2:24-cv-0325-JRG (E.D. Tex.)
EX1017	M&A Complaint	Doc. No. 1, Complaint for Patent Infringement, <i>Autoscribe Corporation v. M&A Ventures, LLC</i> , No. 1:24-cv-04282-SCJ (N.D. Ga.)

Exhibit	Short Name	Description
EX1018	Carlson	U.S. Patent Pub. No. 2008/0319869A1
EX1019	Dierks	U.S. Pat. No. 7,953,642
EX1020	Agrawal	U.S. Pat. No. 9,324,098
EX1021	Collison	U.S. Pat. No. 9,830,596
EX1022	HTML 4.01	HTML 4.01 Specification, Chapter 16: Frames, Dec. 24, 1999, https://www.w3.org/TR/html401/present/frames.html#h-16.5
EX1023	Gage	U.S. Pat. No. 8,543,644
EX1024	PCI	Payment Card Industry (PCI) Data Security Standard – Requirements and Security Assessment Procedures Version 2.0 October 2010
EX1025	Klein	U.S. Patent Pub. No. 2012/0185561A1
EX1026	Waiver of Service	Doc. No. 9, Defendant Nuvei’s Unopposed Motion for Extension of Time to Move, Answer or Otherwise Respond to Plaintiff’s Complaint and Waiver of Summons and of Service Requirements, <i>Autoscribe Corporation v. Nuvei Corporation, et al.</i> , No. 2:24-cv-0325-JRG (E.D. Tex.)
EX1027	Grecia	U.S. Patent Pub. No. 2010/0185868A1
EX1028	Chambers	U.S. Patent Pub. No. 2012/0324225A1
EX1029	Palgon	U.S. Pat. No. 8,458,487

CLAIM LISTING

Claim	Limitation #	Limitation
1	[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:
	[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:
	[1.1.1]	provides financial account registration and token retrieval functions that can be executed to process the payment transaction;
	[1.1.2]	provides access to the financial account registration and token retrieval functions to the merchant server;
	[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;
	[1.1.4]	authenticates the payee; and
	[1.1.5]	executes the financial account registration function, upon initiation by the merchant server, by:
	[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:
	[1.1.5.1a]	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;
	[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;

Claim	Limitation #	Limitation
	[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
	[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;
	[1.2]	receiving the sensitive financial account information provided by the payer via the secure socket layer connection;
	[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;
	[1.4]	executing a token retrieval function, upon initiation by the merchant server via the API, by:
	[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
	[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.
2	[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;

Claim	Limitation #	Limitation
	[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.
3	[3.1]	The method of claim 2, wherein the representation of the non-sensitive electronic data token is the non-sensitive electronic data token.
4	[4.1]	The method of claim 1, wherein the non-sensitive electronic data token constitutes a symbolic representation of the sensitive financial account information.
5	[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.
6	[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.
7	[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.
8	[8.Pre]	A secure server that processes a payment transaction from a payer to a payee, the secure server comprising:
	[8.1]	one or more processors; and
	[8.2]	memory storing instructions that, when executed by the one or more processors, cause the secure server to:
	[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:
	[8.2.1.1]	provide financial account registration and token retrieval functions that can be executed to process the payment transaction;
	[8.2.1.2]	provide access to the financial account registration and token retrieval functions to the merchant server;
	[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;
	[8.2.1.4]	authenticate the payee;

Claim	Limitation #	Limitation
	[8.2.1.5]	execute the financial account registration function, upon initiation by the merchant server, by:
	[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:
	[8.2.1.5.1a]	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;
	[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;
	[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
	[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;
	[8.2.1.6]	receive the sensitive financial account information provided by the payer via the secure socket layer connection;
	[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;
	[8.2.1.8]	execute a token retrieval function, upon initiation by the merchant server by the API, by:

Claim	Limitation #	Limitation
	[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
	[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.
9	[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;
	[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.
10	[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.
11	[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.
12	[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.
13	[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.

Claim	Limitation #	Limitation
14	[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.
15	[15.Pre]	A method of enrolling a payer by a merchant server operated by or for the benefit of a payee, the method comprising:
	[15.1]	providing a webpage to a payer computing system used by a payer
	[15.2]	receiving enrollment data identifying the payer;
	[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;
	[15.4]	displaying a window or frame within the webpage provided to the payer computing system;
	[15.5]	providing authentication credentials to the API to establish an authenticated session, the authentication credentials identifying the payee and authenticating the payee;
	[15.6]	initiating a financial account registration function of the API during the authenticated session, the financial account registration function:
	[15.6.1]	establishing a secure socket layer connection between the secure server and the payer computing system within the window or frame;
	[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
	[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;

Claim	Limitation #	Limitation
	[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;
	[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;
	[15.9]	storing the non-sensitive electronic data token in association with the enrollment data identifying the payer; and
	[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.
16	[16.1]	The method of claim 15, wherein the representation of the non-sensitive electronic data token is the non-sensitive electronic data token.
17	[17.1]	The method of claim 15, wherein the non-sensitive electronic data token constitutes a symbolic representation of the sensitive financial account information.
18	[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.
19	[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).
20	[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.

Claim	Limitation #	Limitation
21	[21.1]	The method of claim 15, wherein the enrollment data identifying the payer is received from the payer over a network.
22	[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.
23	[23.Pre]	A merchant server, operated by or for the benefit of a payee, comprising:
	[23.1]	one or more computer processors; and
	[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:
	[23.2.1]	provide a webpage to a payer computing system used by a payer;
	[23.2.2]	receive enrollment data identifying the payer;
	[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;
	[23.2.4]	display a window or frame within the webpage provided to the payer computing system;
	[23.2.5]	provide authentication credentials to the API to establish an authenticated session, the authentication credentials identifying the payee and authenticating the payee;
	[23.2.6]	initiate a financial account registration function of the API during the authenticated session, the financial account registration function:
	[23.2.6.1]	establishing a secure socket layer connection between the secure server and the payer computing system within the window or frame;
	[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

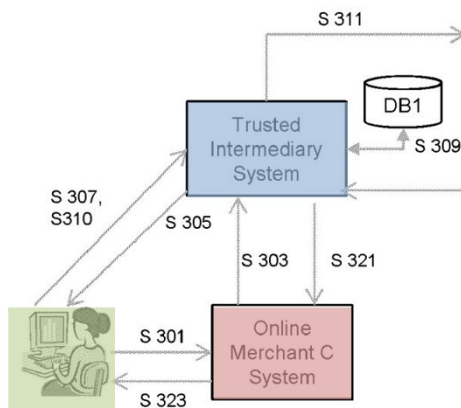
Claim	Limitation #	Limitation
	[23.2.6.3]	outputting instructions to the payer computing system to encrypt the sensitive financial account information provided by the payer and transmit the encrypted financial account information to the secure server via the secure socket layer connection;
	[23.2.7]	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;
	[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;
	[23.2.9]	store the non-sensitive electronic data token in association with the enrollment data identifying the payer; and
	[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.
24	[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.
25	[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:
	[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:

Claim	Limitation #	Limitation
	[25.1.1]	provides financial account registration and token retrieval functions that can be executed to process the payment transaction;
	[25.1.2]	provides access to the financial account registration and token retrieval functions to the merchant server;
	[23.1.3]	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;
	[25.1.4]	authenticates the payee; and
	[25.1.5]	executes the financial account registration function, upon initiation by the merchant server, by:
	[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:
	[23.1.5.1a]	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;
	[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;
	[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
	[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;
	[25.2]	receiving the sensitive financial account information provided by the payer via the secure socket layer connection;

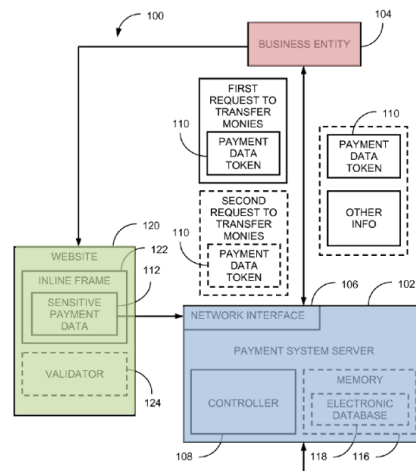
Claim	Limitation #	Limitation
	[25.3]	storing the sensitive financial account information in a secure storage location and performing each software process required to maintain compliance with one or more information security standards;
	[25.4]	executing a token retrieval function, upon initiation by the merchant server via the API, by:
	[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
	[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.
26	[26.1]	The method of claim 25, wherein the URL is a redirect URL that redirects the payer computing system to the secure server.
27	[27.1]	The method of claim 25, wherein the non-sensitive electronic data token constitutes a symbolic representation of the sensitive financial account information.

I. INTRODUCTION

The Challenged Claims are directed to payment processing apparatuses and methods in which a [green] “payer computer system” displays a “webpage provided by the [red] merchant server” along with “a financial account registration form” provided by the [blue] “secure server” “within the window or frame” of “the webpage provided by the merchant server.” *E.g.*, EX1001, Cl. 1. The secure server, rather than the merchant server, obtains and secures sensitive financial information, “thus reducing compliance [and] development costs” for the merchant server. *Id.*, 5:54-6:4. Prior to the ’621 Patent, this claimed architecture and resulting advantages were well known and disclosed in both Stringfellow and Kloster.



Stringfellow (EX1007), FIG. 3



Kloster (EX1008), Fig. 1

As shown, Stringfellow presents the [green] “user” (claimed “payer computer system”) with a composite web page from a [red] “online merchant system” (claimed “merchant server”) along with a financial information collection form beginning

with a “sign in page” from a [blue] “trusted intermediary system” (claimed “secure server”). EX1007, Abstract, FIG. 3. The “sign-in page” “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.” *Id.*, [0075].

Similarly, in Kloster, a [green] customer (claimed “payer computer system”) presenting a single composite web page, “web page 120,” from [red] “business entity 104” (claimed “merchant server”) with a financial information collection form from a [blue] “payment system server 102” (claimed “one or more secure servers”). 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.” EX1008, [0016].

The Challenged Claims include the “secure server” providing the “merchant server” with an “application programming interface (API)” including two API functions. *E.g.*, EX1001, Cl. 1. First, a “financial account registration function” in which the “secure server” “generat[es] a uniform resource locator (URL)” —either “a dynamic URL generated by the secure server for the payer and the payee” or “a static URL and a hypertext transport protocol parameter used by the secure server to identify the payer and the payee.” *Id.* When that URL is received from the “payer

computer system,” it causes the secure server to “output[] instructions to the payer computer system” “to render a financial account registration form” and receive the user’s “sensitive financial information.” *Id.* Stringfellow discloses this function. “[T]he online merchant system” (claimed “merchant server”) “transmit[s] an originating payment authorisation request message to the trusted intermediary system” (claimed “secure server”), which results in “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, [0045], [0075].

Second, the Challenged Claims include a “token retrieval” function in which the “secure server” “provides” to “the merchant server” “a non-sensitive electronic data token representing the sensitive financial information” and “process[es] the payment transaction.” *E.g.*, EX1001, Cl. 1. Kloster discloses this function: after “sensitive payment data” “is collected via the inline frame” by the “payment system server” (the claimed “secure server”), “a payment data token is associated with the sensitive payment data” and the “payment data token is provided to the business entity” (the claimed “merchant server”). EX1008, [0023].

Kloster further provides explicit motivation to add this token functionality to Stringfellow, explaining that “[b]y 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.”* *Id.*, [0021]. This combination renders all Challenged Claims obvious.

II. MANDATORY NOTICES UNDER 37 C.F.R. §42.8

A. Real Party in Interest (37 C.F.R. §42.8(b)(1))

The real parties in interest are Nuvei Technologies, Inc (Petitioner), Nuvei Corporation, and Nuvei Technologies Corporation.

B. Related Matters (37 C.F.R. §42.8(b)(2))

1. *Litigation*

Petitioner is aware of the following lawsuits or related matters involving the '621 Patent:

Case Heading	Number	Court	Filed
<i>Autoscribe Corporation v. Repay Holdings Corp. et al</i>	<i>2-23-cv-00349</i>	E.D. TX	July, 27, 2023
<i>Autoscribe Corporation v. Fortis Payment Systems, LLC</i>	<i>2-23-cv-00364</i>	E.D. TX	Aug. 10, 2023
<i>Autoscribe Corporation v. Payment Savvy, LLC</i>	<i>2-23-cv-00390</i>	E.D. TX	Aug. 29, 2023
<i>Autoscribe Corporation v. Tsevo, LLC et al</i>	<i>3-24-cv-00076</i>	S.D. TX	Mar. 18, 2024
<i>Autoscribe Corporation v. PCI Booking Limited</i>	<i>3-24-cv-00081</i>	S.D. TX	Mar. 22, 2024
<i>Autoscribe Corporation v. Paysafe Limited et al</i>	<i>3-24-cv-00104</i>	S.D. TX	Apr. 11, 2024
<i>Autoscribe Corporation v. Nuvei Corporation et al</i>	<i>2-24-cv-00325</i>	E.D. TX	May 3, 2024

2. *Administrative Proceedings*

The '621 Patent is subject to a pending IPR Petition IPR2024-01159 filed by M&A Ventures, LLC, Repay Holdings Corporation, and Repay Holdings, LLC (collectively, the "IPR2024-01159 Petitioners"). Petitioner Nuvei is neither in privity nor a real party in interest with any of M&A Ventures, LLC, Repay Holdings Corporation, or Repay Holdings, LLC .

C. Designation of Counsel (37 C.F.R. §42.8(b)(3))

Petitioners provide the following designation of counsel.

Lead Counsel	Backup Counsel
Eagle H. Robinson Reg. No. 61,361	Daniel S. Leventhal Reg. No. 59,576

D. Service Information (37 C.F.R. §42.8(b)(4))

Lead Counsel	Backup Counsel
Eagle H. Robinson Norton Rose Fulbright US LLP 98 San Jacinto Blvd., Ste. 1100 Austin, TX 78746 512.536.3083 (telephone) 512.536.4598 (facsimile) eagle.robinson@nortonrosefulbright.com	Daniel S. Leventhal Norton Rose Fulbright US LLP 1550 Lamar Street, Ste. 2000 Houston, TX 77010-4106 713.651.8360 (telephone) 713.651.5246 (facsimile) daniel.leventhal@nortonrosefulbright.com

Petitioner consents to electronic service.

III. GROUNDS FOR STANDING

Petitioner certifies that the '621 Patent is available for IPR, and that Petitioner is not barred or estopped from requesting IPR.

IV. THE GROUNDS IN THIS PETITION ARE NOT CUMULATIVE

A. Original Prosecution

The Examiner did not consider the combination of Stringfellow and Kloster presented here. Section VIII.B. The Examiner did not consider Kloster. *Id.* The Examiner issued rejections in a grandparent application, the '545 Application, that relied upon a reference (U.S. Patent App. Pub. 2012/0030066) having similar disclosure to Stringfellow, but allowed those claims by pointing to a feature —“***a first computer system (e.g. a merchant system) ... collects a communications link to contact the payer, and then sends this information to a second computer system***” —not found in the Challenged Claims. EX1004, 12/387 (“This [feature] is considered to correspond to the features in exemplary claim 1.”); *id.*, 129/387 (“As part of the personal information, a communications channel is provided by the customer and is identified by the system.”).

Because the Challenged Claims do not include the limitation the Examiner found sufficient to allow other claims over Stringfellow, the Examiner did not provide any reasoning disputing Petitioner’s mapping of Stringfellow presented here. As set forth below, the Examiner erred by allowing the claims over the Grounds in this Petition, and the lack of analysis during prosecution weighs against

discretionary denial. *Advanced Bionics, LLC v. MED-EL Elektromedizinische Geräte Ges.m.b.H et al.*, No. IPR2019-01469, Paper 6 at 10-11 (P.T.A.B. Feb. 13, 2020).

B. IPR2024-01159

Discretionary denial, either under *General Plastics* or *Advanced Bionics* in view of IPR2024-01159 is not appropriate. First, Petitioner is not the same, not an RPI of, a co-defendant with, nor accused of infringing based on the same products as any of the IPR2024-01159 Petitioners. *Compare* EX1016, 5-6/28 *with* EX1017, 6/22; *see also* *Valve Corp. v. Elec. Scripting Prods., Inc.*, IPR2019-00062, Paper 11 at 9-10. “[W]here, as here, the first and second petitioners are neither the same party, nor possess a significant relationship under *Valve*, *General Plastic* factor one necessarily outweighs the other *General Plastic* factors.” *Videndum Prod. Solutions, Inc. v. Rotolight Ltd.*, IPR2023-01218, Paper 12 (Director Review) at 5-6. Second, addressing this Petition would not be an inefficient use of the Board’s resources or interfere with the Board’s decision in IPR2024-01159. This Petition relies on different art than IPR2024-01159, which does not reference Stringfellow and cites to Kloster only as a background reference. This Petition also challenges additional claims—Claims 11, 13-14, 20, and 26-27—not challenged in IPR2024-01159. Further, IPR2024-01159 proposes three claim constructions and appears to rely on those constructions in mapping its art to the claims. Section IX. This Petition’s

mapping does not require any of those constructions. *Id.* Therefore, *General Plastics* factors 2, 6, and 7 weigh against discretionary denial. Finally, Petitioner has not delayed in filing this Petition. Patent Owner filed this suit in May of this year and, having not been served, Petitioner waived service in July. *See* EX1016; EX1026.

V. STATEMENT OF PRECISE RELIEF REQUESTED FOR EACH CLAIM CHALLENGED

A. Claims for Which Review Is Requested (37 C.F.R. § 42.104(b)(1))

Petitioner requests review and cancellation of Claims 1-27.

B. Statutory Grounds of Challenge (§42.104(b)(2))¹

Ground 1: Claims 1-27 are unpatentable under 35 U.S.C. §103 over U.S. Publication No. 2010/0174626 (“Stringfellow”) in view of U.S. Publication No. 2010/0094755 (“Kloster”). Stringfellow was published on July 8, 2010 and is, therefore, prior art under at least pre-AIA 35 U.S.C. §102(b). Kloster was published on April 15, 2010 and is, therefore, prior art under at least pre-AIA 35 U.S.C. §102(e).

Ground 2: Claims 1-27 are unpatentable under 35 U.S.C. §103 over Stringfellow in view of Kloster and U.S. Patent Publication No. 2008/0319869A1

¹ All Grounds are supported by a POSITA’s general knowledge. *Koninklijke Philips N.V. v. Google LLC*, 948 F.3d 1330, 1337-38 (Fed. Cir. 2020).

(“Carlson”). Carlson was filed June 20, 2008 and published December 25, 2008, and is, therefore, prior art under at least pre-AIA 35 U.S.C. §102(b).

VI. LEVEL OF ORDINARY SKILL IN THE ART

A person of ordinary skill in the art (“POSITA”) pertinent to the ’621 Patent would have had at least (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. EX1005 ¶36. The prior art also evidence this level of ordinary skill. *Id.* Here, the background technology described in Section VII and the prior art described in Section X demonstrate that a POSITA would have been familiar with the underlying principles of 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. *Id.* 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. *Id.* Additional education might substitute for some experience, and vice versa. *Id.*

VII. FIELD OF TECHNOLOGY

A. HTTP communications

1. *HTTP Communications*

Web communications were well-known prior to the priority date of the '621 Patent. EX1005, ¶37; EX1011, 1/176. 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. URLs can contain a query string, starting with a “?” symbol, that includes parameters providing further instructions to the server. *Id.*

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

HTTP is a stateless protocol, which means that an HTTP server treats each request it receives independently without knowledge of any past history. EX1005, ¶39. 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. *Id.* By 2012, it was well known to convey such a context by including information in the query string of a URL in an HTTP request. EX1005, ¶39. For example, Carlson discloses a URL “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, [0100], [0063], [0042]; *see also* EX1013, 9:21-30 (URL including in query string “information as may be needed to sell an item”); EX1023, 5:32-43.

3. *HTML and iFrames*

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; EX1005, ¶40; EX1007, [0075]; EX1008, [0016].

It was also known to load an iFrame with a URL containing state specific query string parameters:

```
<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>
```

EX1018, [0100] (“<iframe ‘src=‘http://www.ivisa.com/zifpvisa/
php?Merchid=2000Scorrid=123’”); EX1005, ¶41; EX1013, 9:21-30.

4. *HTTPS / Secure Socket Layer (“SSL”) Connections*

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)*

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. EX1005, ¶43; *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. EX1005, ¶43; *see e.g.*, EX1007, [0072]-[0073]; EX1008, [0015]; EX1014, 111:61-63; 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.”); EX1005, ¶43. “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]; EX1005, ¶43.

B. Online Shopping and Payments

1. Online Shopping and Payment Processing

By 2012, eCommerce was well known, as was online payments. EX1007, [0003]; EX1005, ¶44. 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; EX1005, ¶44. 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; EX1005, ¶44; EX1012, 1/67.

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; EX1005 ¶45; 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. EX1005, ¶45.

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].

To provide a more seamless experience to the user, it was well known for eCommerce sites to use 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; EX1005, ¶46. 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. EX1005, ¶46; EX1007, [0075]; EX1018, [0086]; EX1020, 3:51-55; EX1019, 2:28-60.

2. *Tokenization*

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* Section VII.B.1. 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. 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]; EX1005, ¶47. This approach enhanced security by reducing the risk of sensitive data exposure while allowing merchants to handle transactions smoothly. *Id.* 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. THE '621 PATENT

A. Overview

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; *id.*, 5:62-6:4. A “[s]ecure server 202...stores the financial account information received from the customer.” *Id.*, 7:31-34; EX1005, ¶48.

The merchant server initiates two separate and distinct functions during the payment process: the financial account registration function and the token retrieval

function. EX1001, 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. EX1005, ¶49. Through that communication link, the payer can submit her sensitive financial account information to the secure server to be stored and tokenized. EX1001, 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; EX1005, ¶49.

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)." EX1001, 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; EX1005, ¶50.

Third, to process the payment from the payer, the merchant then initiates the token retrieval function. EX1005, ¶51. 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.” EX1001, 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; EX1005, ¶51.

However, as set forth in this Petition, the ’621 Patent merely describes standard payment systems and methods of securing communications of financial information within these systems. EX1005, ¶52. The Board should thus institute review because there is a strong likelihood that Petitioner will prevail on at least one Challenged Claim.

B. Prosecution History

Patent Owner filed Application No. 16/535,424 (the “’424 Application”), which resulted in the ’621 Patent, on August 8, 2019 and was granted an allowance following repeated 101 rejections. EX1002, 2-35/442 (Notice of Allowance), 223/442 (101 rejection), 303/442 (same), 370/442 (same). The Examiner made no prior art rejections and stated in the Notice of Allowability that the subject application was allowable over 35 U.S.C. 101. EX1002, 7/442.

Much of the prosecution history of the '621 Patent lies in the history of its parent applications, U.S. Patent Application No. 15/408,185 (“the '185 Application”) and U.S. Patent Application No. 13/679,545 (“the '545 Application”). EX1003; EX1004. In particular, for the '545 Application, after multiple final rejections, the Examiner in the Notice of Allowance stated that neither Hammad nor Stringfellow taught “a first computer system (e.g. a merchant system), after collecting non-account information identifying a payer, *also collects a communications link to contact the payer, and then sends this information to a second computer system* [e.g., the secure server] which contacts the payer via the communications link and collects the payer's account information from the payer.” EX1004, 12/387.

IX. CLAIM CONSTRUCTION

Petitioner interprets the Challenged Claims according to the *Phillips* claim construction standard. 83 Fed. Reg. 51340, 51340-44 (Oct. 11, 2018); *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005). Construction of any term is not necessary to address the mappings presented in this Petition, and the terms of the Challenged Claims should be given their plain and ordinary meaning for purposes of this proceeding. *See, e.g., Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017).

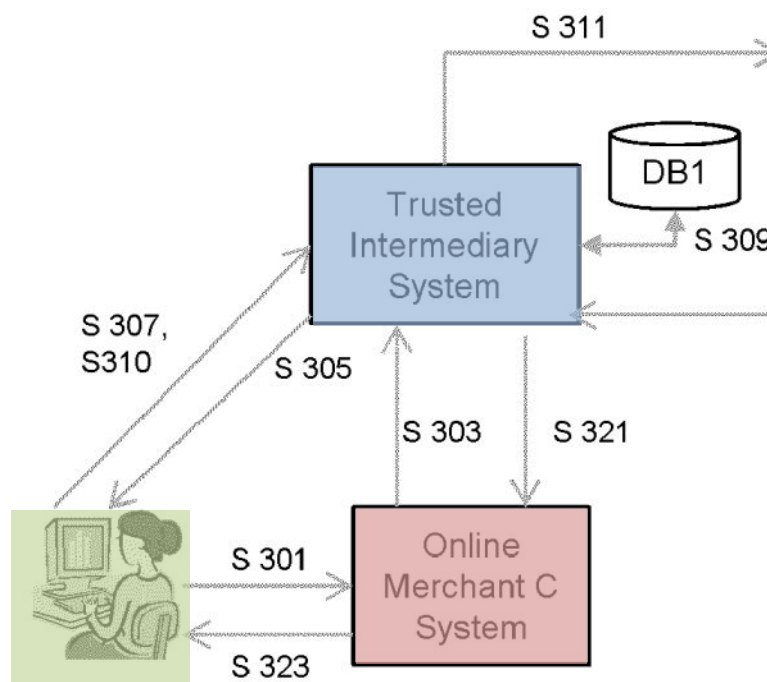
Petitioner notes that IPR2019-00062 proposes construction of three terms and appears to have relied on those constructions in mapping the art. IPR2019-00062, Paper 4, 10-21, 43-44, 66-67. This Petition's mapping does not rely on the distinctions raised by those proposed construction with which Petitioner does not necessarily agree. First, IPR2019-00062 proposes that "without providing the non-sensitive electronic data token to the payer" / "the secure server does not provide the non-sensitive electronic data token to the payer" refer to a period during performance of the "token retrieval function." *Id.*, 11-17. As discussed below, the claimed token is not provided to the payer at any time in the proposed combination and, thus, it is not necessary to specify a time period for those limitations. Limitation 1.4.1. Second, IPR2019-00062 proposes that the "secure server" "generating" a URL "comprising" "a static URL and a ... HTTP parameter" "does not strictly require that the URL be generated entirely at the secure server" and can be met by the secure server sending a parameter that "prompts the merchant server to combine that [parameter] with a static URL." *Id.*, 17-21, 44. While this interpretation is certainly consistent with Patent Owner's apparent infringement theory, the interpretation is inconsistent with the plain claim language. Regardless, as discussed below, Stringfellow discloses the secure server generating a URL and, therefore, this construction need not be reached for this Petition.

X. REASONS FOR THE RELIEF REQUESTED (§§42.22(a)(2), 42.104(b)(4))

A. Ground 1: Stringfellow and Kloster Render Obvious Claims 1-27

1. *Stringfellow*

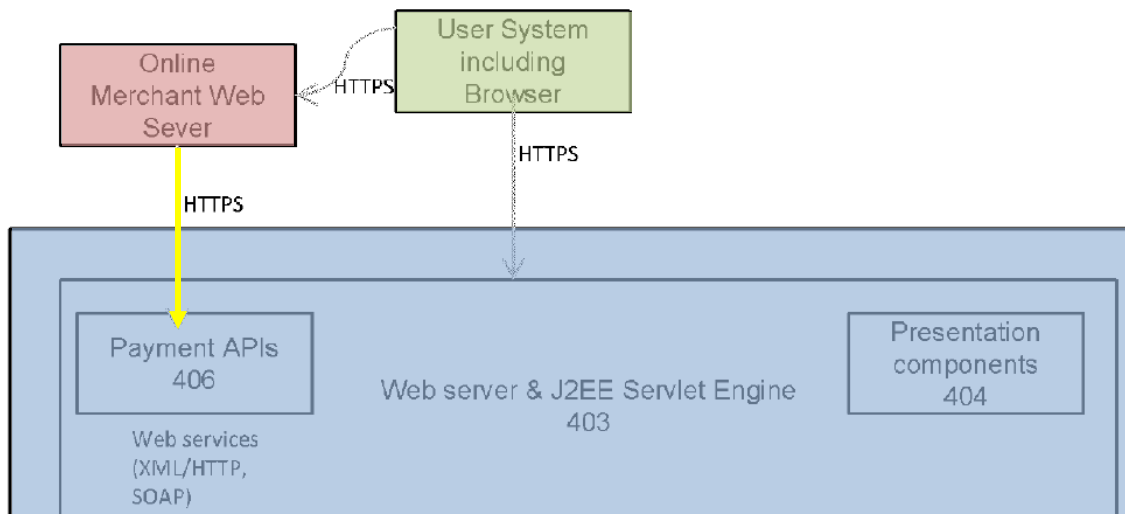
Stringfellow discloses “a method of processing payment authorization requests for payment transactions.” EX1007, Abstract. The “user” (green, meets the claimed “payer computer system”) is served by a first server, “online merchant system” (red, meets the claimed “merchant server”) providing an online 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).

“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]; EX1005, ¶54. 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. EX1005, ¶54.

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. EX1005, ¶55.



EX1007, FIG. 2 (excerpted, annotated).

**a) Online Merchant System → Trusted Intermediary:
Originating Payment Authorisation Request Message.**

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]; EX1005, ¶56. 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]; EX1005, ¶56. 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].

**b) Trusted Intermediary → Online Merchant System:
Generation and Retrieval Of Sign-In Page URL**

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]; EX1005, ¶57.²

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

“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

² 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, a POSITA 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. EX1005, ¶58.

web principles, this is accomplished by: (a) the online merchant system sending to the client browser code that instructs the browser to send, to the trusted intermediary system, an HTTP request that specifies the sign-in URL, and (b) the trusted intermediary system sending, to the client browser, an HTTP response with instructions to display the sign-in page. EX1005, ¶59.

“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]. A POSITA would understand, therefore, that 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). EX1005, ¶60.

**d) Client ↔ Trusted Intermediary: Account Registration,
Including Collecting and Storing Sensitive Financial
Information**

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 would provide the registration screens for display within that same iframe. EX1005, ¶61.

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. EX1005, ¶62.

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

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].

e) Content of “Sign-In URL”

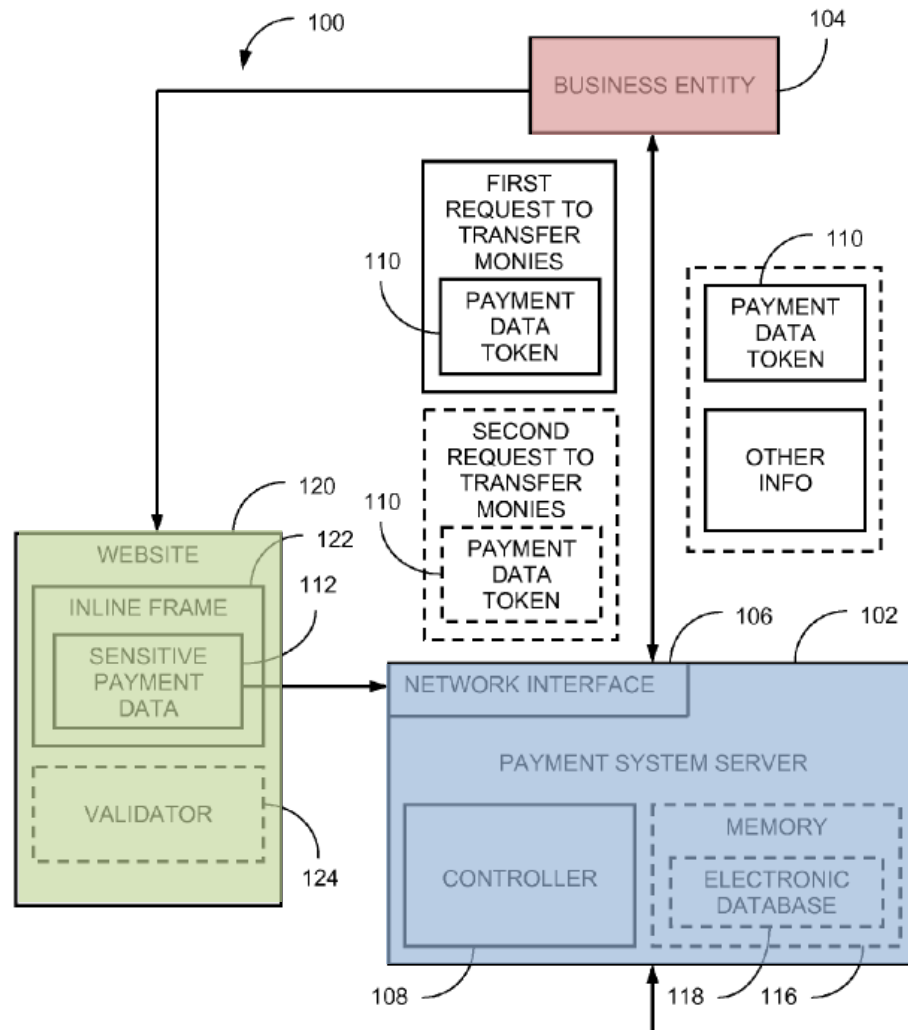
A POSITA would understand that the sign-in URL that the trusted intermediary system provides to the online merchant system in response to an originating payment authorization request message includes information linking the URL to that request message. EX1005, ¶65. This is necessarily true because Stringfellow 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 a specific 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]; EX1005, ¶65.

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. EX1005, ¶66. Accordingly, a POSITA would have understood Stringfellow to disclose or render obvious 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. *Id.* As evidence that this approach was within the knowledge of 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]; EX1005, ¶66. 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]. 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. EX1005, ¶66; EX1018, [0080], [0100].

2. *Kloster*

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.



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

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; EX1005, ¶68. “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].

3. Motivation to Combine

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. EX1005, ¶69. As discussed in Sections X.A.1-X.A.2, 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. EX1005, ¶69. Thus, a POSITA would look to Kloster both as providing additional compatible

functionality and additional implementation details for implementing common functionality. *Id.*

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. EX1005, ¶70; 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”).

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. EX1005, ¶71.

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. EX1005, ¶72. 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. EX1005, ¶72.

Kloster also describes additional 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. EX1005, ¶73.

4. Modification of Stringfellow in view of Kloster

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. EX1005, ¶74. As discussed in Sections X.A.1-X.A.3, 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. EX1005, ¶74.

With those functions integrated, tokens can advantageously be used in multiple payment flows within Stringfellow. First, as discussed in Section X.A.1, 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 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. EX1005, ¶75.

Applying Kloster’s disclosure, Stringfellow’s online merchant system may obtain a payment token after “sensitive payment data” “is collected” in multiple ways. EX1005, ¶76. “[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]; EX1005, ¶76. 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 X.A.1, 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. EX1005, ¶76.

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]; EX1005, ¶77. Finally, “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 is 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.” EX1005, ¶77.

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. EX1005, ¶78.

5. *Element-by-Element Analysis*

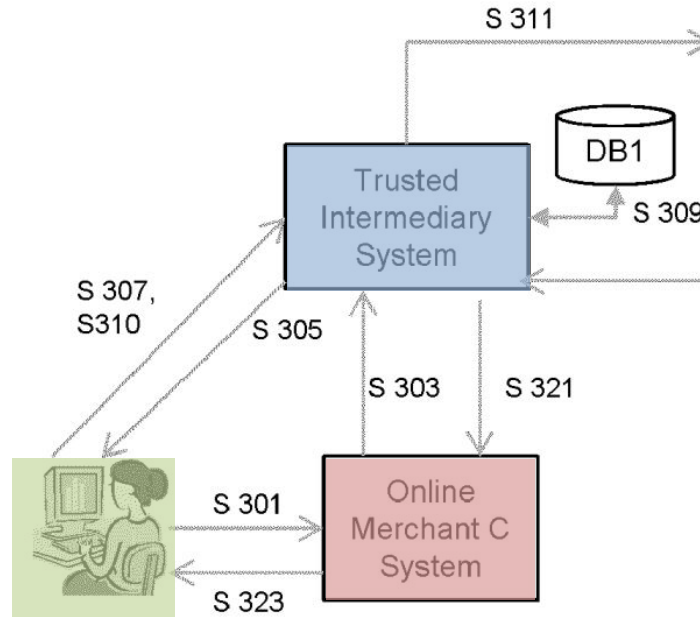
The analysis of each element below incorporates the above descriptions of each reference, the motivation to combine the references, and the resulting modification.

a. Claim 1

[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:

To the extent the preamble may be limiting, Stringfellow/Kloster renders obvious Limitation [1.pre]. EX1005, ¶80. As discussed in X.A.1 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]; X.A.4.



EX1007, Fig. 3 (excerpt, annotated).

[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:

Stringfellow/Kloster renders obvious Limitation [1.1]. EX1005, ¶¶81-82. As discussed in X.A.1, 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.

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]; EX1005, ¶82.

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

Stringfellow/Kloster renders obvious Limitation [1.1.1]. EX1005, ¶¶83-84. As discussed in d), 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; EX1005, ¶83.

Further, as discussed in X.A.4 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; EX1005, ¶84. *See also* Limitations [1.1.5]-[1.1.5.4], [1.4]-[1.4.2].

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

Stringfellow/Kloster renders obvious Limitation [1.1.2]. As discussed in X.A.1 and X.A.4, 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]. A POSITA would understand that the Stringfellow/Kloster combination provides access through API configurations that allow the merchant server to initiate financial account registration and tokenization functions for processing payment transactions. EX1005, ¶85. *See also* Limitation [1.1.1]

[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;

Stringfellow/Kloster renders obvious Limitation [1.1.3]. EX1005, ¶86. As discussed in 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]. A POSITA would understand that “an identifier for the order” is necessarily associated with the payer that created the order. EX1007, [0049]; EX1008, [0015]; EX1005, ¶86.

[1.1.4] authenticates the payee; and

Stringfellow/Kloster renders obvious Limitation [1.1.4]. EX1005, ¶87. 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. EX1005, ¶87. 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. EX1005, ¶87. 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; EX1005, ¶87. 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. EX1005, ¶87.

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”), the nature of the application, and the known role of use of certificates to perform authentication. *Id.*, 48/67, EX1005, ¶87; VII.A.5.

Second, as discussed in X.A.1, 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]. EX1005, ¶87. 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]; EX1005, ¶87. The use of such a merchant ID to perform authentication is consistent with known API authentication techniques as discussed in VII.A.5. EX1005, ¶87.

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

Stringfellow/Kloster renders obvious Limitation [1.1.5]. As discussed in 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; EX1005, ¶88. *See also* Limitations [1.1.5.1]-[1.1.5.4].

[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:

Stringfellow/Kloster renders obvious Limitation [1.1.5.1]. EX1005, ¶89. As discussed in b)-c), 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]; EX1005, ¶89. A POSITA would understand that to provide a sign in URL linked to the request message as discussed in X.A.1, this trusted intermediary system must generate that URL. EX1005, ¶89. After providing the sign-in page URL, the trusted intermediary system “subsequently...[provides for] the ***creation of a secure session.***” EX1007, [0075] (emphasis added); EX1005, ¶89. 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; EX1005, ¶89. A POSITA would understand that the URL provides for creating “secure session” with the user based on an “HTTP

over Secure Socket Layer protocol (HTTPS).” EX1005, ¶89. *See also* Limitation [1.1.5.1a].

[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;

Stringfellow/Kloster renders obvious Limitation [1.1.5.1a]. EX1005, ¶90. As explained in e), 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]; EX1005, ¶90. 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. EX1005, ¶90; *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. EX1005, ¶90. Regardless of whether including parameters in a query string of a URL is considered generating a

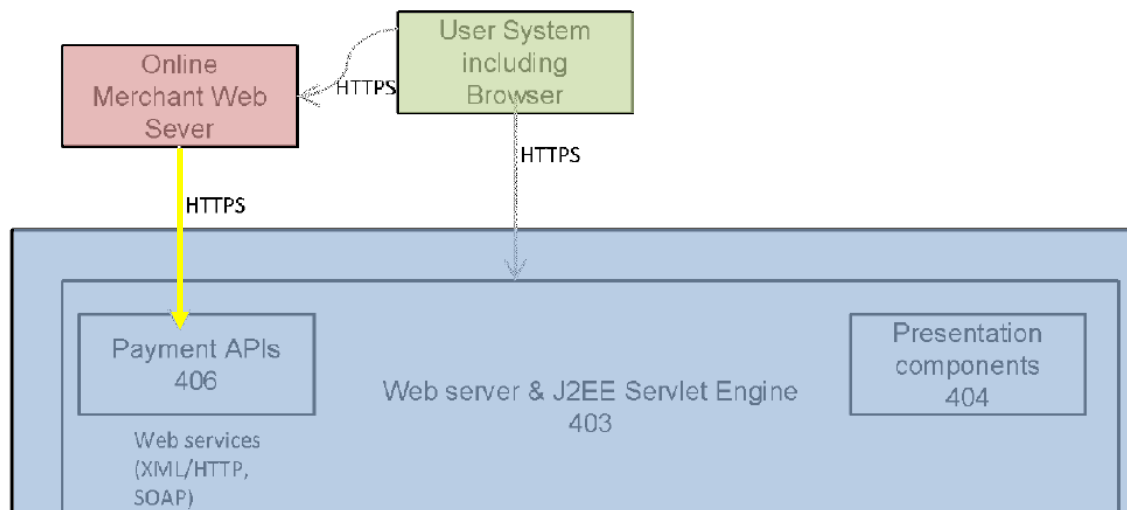
“dynamic” URL or “a static URL and a hypertext transport protocol (HTTP) parameter,” the trusted intermediary’s generation of the sign-in URL meets at least one of the recited options. *Id.*

[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;

Stringfellow/Kloster renders obvious Limitation [1.1.5.2]. As discussed in c), in response to the merchant server’s payment authorization 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); EX1005, ¶91. 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]; EX1005, ¶91. A POSITA would understand that once the iFrame is displayed to the user, the user’s browser would detect the HTTPS protocol in the iFrame’s src attribute and then initiate the “secure session” through an “initial handshake” with the trusted intermediary system as discussed in VII.A.4. EX1005, ¶91. 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; EX1005, ¶91.

Further, a POSITA would recognize that the merchant server’s “payment authorization request message” is 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; EX1005, ¶92.



EX1007, FIG. 2 (excerpted, annotated).

[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

Stringfellow/Kloster renders obvious Limitation [1.1.5.3]. EX1005, ¶93. As discussed in discussed in c), 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 discussed in VII.A.3, a POSITA would understand that 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. EX1005, ¶93.

[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;

Stringfellow/Kloster renders obvious Limitation 1.1.5.4]. EX1005, ¶94. As discussed in X.A.1, 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 VII.A.4, 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.” EX1005, ¶94. Upon receipt of the HTTPS request, the trusted intermediary system sends an HTTPS response that contains additional instructions for the payer computer system, including the 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; EX1005, ¶94. 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. EX1005, ¶94. 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]; EX1005, ¶94. *See also* Limitation [1.2].

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

Stringfellow/Kloster renders obvious Limitation [1.2]. EX1005, ¶95. 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].

[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;

Stringfellow/Kloster renders obvious Limitation [1.3]. EX1005, ¶¶96-98. 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].

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); EX1005, ¶97.

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. EX1005, ¶98. Kloster explains that “collecting such data [e.g., credit card numbers] may subject the online merchant to

costly security audits and regulations.” EX1008, [0003], [0028]; *see also* EX1021 1:45-60 (“storage of their customers’ sensitive payment information...must also be in compliance with the credit card networks as defined by various standards such as the PCI Security Standard.”); EX1005, ¶98. Similarly, the ’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.

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

Stringfellow/Kloster renders obvious Limitation [1.4]. EX1005, ¶99. As discussed in X.A.4, token retrieval may occur where 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]; EX1005, ¶99. 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. EX1005, ¶99; *see also* EX1025, [0005], [0051]-[0055].

[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

Stringfellow/Kloster renders obvious Limitation [1.4.1]. EX1005, ¶100. 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 in X.A.4 and 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]; EX1005, ¶100. 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]; EX1005, ¶100. 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 security³ by unnecessarily exposing a token that can be used to request payments. EX1005, ¶100.⁴

[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.

Stringfellow/Kloster renders obvious Limitation [1.4.2]. EX1005, ¶101. As discussed in X.A.4, in the Stingfellow/Kloster combination the trusted intermediary system/payment system processes 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

³ For example, a bad actor obtaining the token may attempt to combine with other user data to request payment. EX1005, ¶100.

⁴ 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. EX1005, ¶100.

end user to the business entity utilizing the retrieved sensitive payment data via a backend payment gateway.” EX1008, [0004], [0015], [0021], [0023], [0024].

b. Claim 2

[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;

Stringfellow/Kloster renders obvious Limitation [2.1] for reasons discussed above in Limitations [1.4.1], [1.4.2], and as discussed in X.A.2 and X.A.4. EX1005, ¶102. 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].

[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.

Stringfellow/Kloster renders obvious Limitation [2.2] for reasons discussed above in Limitation [2.1]. EX1005, ¶103. 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].

c. Claim 3

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

Stringfellow/Kloster renders obvious Limitation [3.1] for reasons discussed above in Limitation [2.1]. EX1005, ¶104.

d. Claim 4

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

Stringfellow/Kloster renders obvious Limitation [4.1] for reasons discussed above in Limitation [1.4.1]. EX1005, ¶105.

e. Claim 5

[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.

Stringfellow/Kloster renders obvious Limitation [5.1] for reasons discussed above in Limitation [1.1.3]. EX1005, ¶106. 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. EX1005, ¶106. 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.” EX1005, ¶106.

f. Claim 6

[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.

Stringfellow/Kloster renders obvious Limitation [6.1]. EX1005, ¶107.

First, as discussed for Limitation [1.1.5.2], Stringfellow/Kloster renders obvious providing a secure socket layer connection between the secure server and the payer computer system.

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). EX1005, ¶108. 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].

As discussed in X.A.1, 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.” EX1005, ¶109; EX1001, 9:36-40, 10:27-30.

g. Claim 7

[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.

Stringfellow/Kloster renders obvious Limitation [7.1] for reasons discussed above in Limitation [1.2]. EX1005, ¶110.

h. Claim 8

[8.Pre]

Stringfellow/Kloster renders obvious Limitation [8.Pre] for reasons discussed above in Limitation [1.Pre] and below in Limitation [8.1]. EX1005, ¶111.

[8.1] one or more processors; and

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], EX1005, ¶112. 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. EX1005, ¶112.

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

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], EX1005, ¶113. 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.” EX1005, ¶113.

[8.2.1]

See Limitation [1.1]. EX1005, ¶114.

[8.2.1.1]

See Limitation [1.1.1]. EX1005, ¶115.

[8.2.1.2]

See Limitation [1.1.2]. EX1005, ¶116.

[8.2.1.3]

See Limitation [1.1.3]. EX1005, ¶117.

[8.2.1.4]

See Limitation [1.1.4]. EX1005, ¶118.

[8.2.1.5]

See Limitation [1.1.5]. EX1005, ¶119.

[8.2.1.5.1]

See Limitation [1.1.5.1]. EX1005, ¶120.

[8.2.1.5.1a]

See Limitation [1.1.5.1a]. EX1005, ¶121.

[8.2.1.5.2]

See Limitation [1.1.5.2]. EX1005, ¶122.

[8.2.1.5.3]

See Limitation [1.1.5.3]. EX1005, ¶123.

[8.2.1.5.4]

See Limitation [1.1.5.4]. EX1005, ¶124.

[8.2.1.6]

See Limitation [1.2]. EX1005, ¶125.

[8.2.1.7]

See Limitation [1.3]. EX1005, ¶126.

[8.2.1.8]

See Limitation [1.4]. EX1005, ¶127.

[8.2.1.8.1]

See Limitation [1.4.1]. EX1005, ¶128.

[8.2.1.8.2]

See Limitation [1.4.2]. EX1005, ¶129.

i. Claim 9

[9.1]

See Limitation [2.1]. EX1005, ¶130. For the server of Claim 8, *see supra*, Limitation [8.1] and, for instructions causing the act, *see supra*, Limitation [8.2].

[9.2]

See Limitation [2.2]. EX1005, ¶¶131-132.

j. Claim 10

[10.1]

See Limitation [3.1]. EX1005, ¶133. For the server of Claim 9, *see supra*, Limitation [8.1].

k. Claim 11

[11.1]

See Limitation [4.1]. EX1005, ¶134. For the server of Claim 8, *see supra*, Limitation [8.1].

l. Claim 12

[12.1]

See Limitation [5.1]. EX1005, ¶135. For the server of Claim 8, *see supra*, Limitation [8.1].

m. Claim 13

[13.1]

See Limitation [6.1]. EX1005, ¶136. For the server of Claim 8, *see supra*, Limitation [8.1].

n. Claim 14

[14.1]

See Limitation [7.1]. EX1005, ¶137. For the server of Claim 8, *see supra*, Limitation [8.1].

o. Claim 15

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

See Limitations [1.Pre], [5.1], and [6.1]. EX1005, ¶138.

[15.1] providing a webpage to a payer computing system used by a payer;

See Limitation [1.1]. EX1005, ¶139.

[15.2] receiving enrollment data identifying the payer;

See Limitations [1.1.3], [5.1] and [6.1]. EX1005, ¶140.

[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;

See Limitation [1.1.3]. EX1005, ¶141.

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

See Limitation [1.1.5.2]. EX1005, ¶142.

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

See Limitations [1.1.3] and [1.1.4]. EX1005, ¶143.

[15.6] initiating a financial account registration function of the API during the authenticated session, the financial account registration function:

See Limitation [1.1.5]. EX1005, ¶144.

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

See Limitation [1.1.5.2]. EX1005, ¶145.

[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

See Limitation [1.1.5.3]. EX1005, ¶146.

[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;

See Limitation [1.1.5.4]. EX1005, ¶147.

[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;

See Limitation [1.4] and [1.4.1]. EX1005, ¶148.

[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;

See Limitation [1.4.1]. EX1005, ¶149.

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

See Limitations [1.4.1], [2.1], and [6.1]. EX1005, ¶150. 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. EX1005, ¶150.

[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.

See Limitations [1.4], [1.4.1], and [1.4.2]. EX1005, ¶151. 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]; EX1005, ¶151. 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]; EX1005, ¶151.

p. Claim 16

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

See Limitation [3.1]. EX1005, ¶152.

q. Claim 17

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

See Limitation [4.1]. EX1005, ¶153. For the method of Claim 15, *see supra*, Limitation [15.Pre].

r. Claim 18

[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.

See Limitation [5.1]. EX1005, ¶154.

s. Claim 19

[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).

See Limitation [1.1.5.1a]. EX1005, ¶155. For the method of Claim 15, *see supra*, Limitation [15.Pre].

t. Claim 20

[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.

See Limitation [6.1]. EX1005 ¶156. For the method of Claim 15, *see supra*, Limitation [15.Pre].

u. Claim 21

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

See Limitation [6.1]. EX1005, ¶157. For the method of Claim 15, *see supra*, Limitation [15.Pre].

v. Claim 22

[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.

See Limitation [7.1]. EX1005, ¶158. For the method of Claim 15, *see supra*, Limitation [15.Pre].

w. Claim 23

[23.Pre]

See Limitation [15.Pre]. EX1005, ¶159.

[23.1]

See Limitation [8.1]. EX1005, ¶160.

[23.2]

See Limitation [8.2]. EX1005, ¶161.

[23.2.1]

See Limitation [15.1]. EX1005, ¶162.

[23.2.2]

See Limitation [15.2]. EX1005, ¶163.

[23.2.3]

See Limitation [15.3]. EX1005, ¶164.

[23.2.4]

See Limitation [15.4]. EX1005, ¶165.

[23.2.5]

See Limitation [15.5]. EX1005, ¶166.

[23.2.6]

See Limitation [15.6]. EX1005, ¶167.

[23.2.6.1]

See Limitation [15.6.1]. EX1005, ¶168.

[23.2.6.2]

See Limitation [15.6.2]. EX1005, ¶169.

[23.2.6.3]

See Limitation [15.6.3]. EX1005, ¶170.

[23.2.7]

See Limitation [15.7]. EX1005, ¶171.

[23.2.8]

See Limitation [15.8]. EX1005, ¶172.

[23.2.9]

See Limitation [15.9]. EX1005, ¶173.

[23.2.10]

See Limitation [15.10]. EX1005, ¶174.

x. Claim 24

[24.1]

See Limitation [4.1]. EX1005, ¶175. For the server of Claim 23, *see supra*, §X.A.5.w.

y. Claim 25

[25.Pre]

See Limitation [1.Pre]. EX1005, ¶176.

[25.1]

See Limitation [1.1]. EX1005, ¶177.

[25.1.1]

See Limitation [1.1.1]. EX1005, ¶178.

[25.1.2]

See Limitation [1.1.2]. EX1005, ¶179.

[25.1.3]

See Limitation [1.1.3]. EX1005, ¶180.

[25.1.4]

See Limitation [1.1.4]. EX1005, ¶181.

[25.1.5]

See Limitation [1.1.5]. EX1005, ¶182.

[25.1.5.1]

See Limitation [1.1.5.1]. EX1005, ¶183.

[25.1.5.1a]

See Limitation [1.1.5.1a]. EX1005, ¶184.

[25.1.5.2]

See Limitation [1.1.5.2]. EX1005, ¶185.

[25.1.5.3]

See Limitation [1.1.5.3]. EX1005, ¶186.

[25.1.5.4]

See Limitation [1.1.5.4]. EX1005, ¶187.

[25.2]

See Limitation [1.2]. EX1005, ¶188.

[25.3]

See Limitation [1.3]. EX1005, ¶189.

[25.4]

See Limitation [1.4]. EX1005, ¶190.

[25.4.1]

See Limitation [1.4.1]. EX1005, ¶191.

[25.4.2]

See Limitation [1.4.2]. EX1005, ¶192.

z. Claim 26

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

See Limitation [1.1.5.1]. EX1005, ¶193. For the method of Claim 25, *see supra*, Limitation [25.Pre].

aa. Claim 27

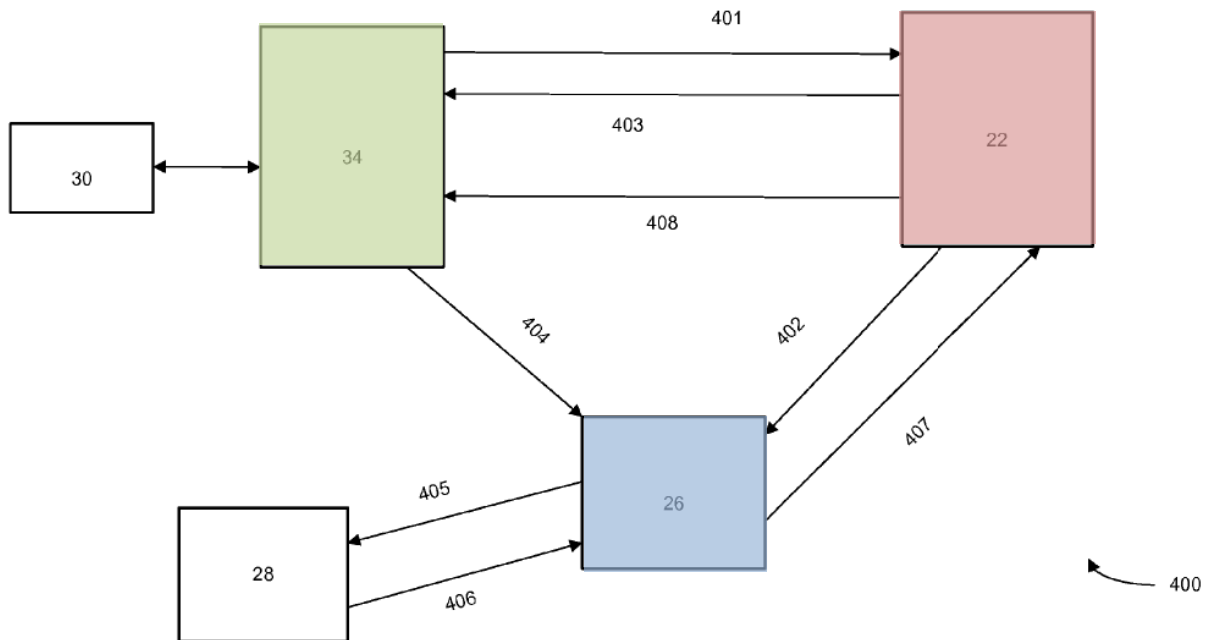
[27.1]

See Limitation [4.1]. EX1005, ¶194. For the method of Claim 25, *see supra*, Limitation [25.Pre].

B. Ground 2: Stringfellow, Kloster, and Carlson Render Obvious Claims 1-27

1. Carlson

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).

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].

2. Motivation to Combine

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]; EX1005, ¶97.

3. Modification of Stringfellow in view of Kloster and Carlson

Applying Carlson’s teachings to the Stringfellow/Kloster combination results in the same system as already described in X.A.1 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].

4. Element-By-Element Analysis

Because 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

Petitioner respectfully requests cancellation of the Challenged Claims.

Date: November 6, 2024

By: /Eagle H. Robinson/
Eagle H. Robinson
Reg. No. 61,361

Lead Counsel for Petitioners

CERTIFICATION UNDER 37 C.F.R. §42.24

Pursuant to 37 C.F.R. § 42.24(a)(i), the undersigned certifies that this Paper—exclusive of the table of contents, mandatory notices under § 42.8, certificate of service, and this certificate of word count—includes 13,294 words.

Date: November 6, 2024

/Eagle H. Robinson/
Eagle H. Robinson (Reg. No. 61,361)

CERTIFICATE OF SERVICE

Pursuant to 37 C.F.R. § 42.6(e) and 37 C.F.R. § 42.105(a), the undersigned certifies that on November 6, 2024, a complete copy of this Petition for Inter Partes Review, Petitioner's Power-of-Attorney, and EX1001-EX1029 were served on Patent Owner at the correspondence addresses of record listed below by USPS Priority Mail Express®:

BLANK ROME LLP
1825 EYE STREET NW
WASHINGTON, DC 20006-5403

Courtesy copies of the same documents are also being sent to litigation counsel for Patent Owners' exclusive licensee, Autoscribe Corporation, via email at:

jmcmanis@azalaw.com
cphillips@azalaw.com
cdeng@azalaw.com
apeterson@azalaw.com
mkillingsworth@azalaw.com
andrea@wsfirm.com

/Eagle H. Robinson/
Eagle H. Robinson (Reg. No. 61,361)