

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

M&A VENTURES, LLC, REPAY HOLDINGS CORPORATION, and REPAY
HOLDINGS, LLC
Petitioners,

v.

AUTOSCRIBE CORPORATION,
Patent Owner.

IPR2024-01159
U.S. Patent No. 11,620,621

**PETITION FOR *INTER PARTES* REVIEW
UNDER 35 U.S.C. § 312 AND 37 C.F.R. § 42.104**

TABLE OF CONTENTS

I.	INTRODUCTION	7
II.	GROUND FOR STANDING.....	8
III.	NOTE.....	8
IV.	SUMMARY OF THE '621 PATENT	8
V.	LEVEL OF ORDINARY SKILL IN THE ART	10
VI.	CLAIM CONSTRUCTION	10
A.	<i>“providing a non-sensitive data token representing the sensitive financial account information to the merchant... without providing the non-sensitive data token to the payer”</i>	11
B.	<i>“receiving the non-sensitive electronic data token from the secure server, wherein...the secure server does not provide the non-sensitive electronic data token to the payer”</i>	15
C.	<i>“generating a uniform resource locator (url), ... the url comprising either: a dynamic url generated by the secure server...; or a static url and a hypertext transport protocol (http) parameter...”</i>	17
VII.	RELIEF REQUESTED	21
VIII.	DISCRETIONARY DENIAL WOULD BE INAPPROPRIATE.....	21
A.	The Fintiv Factors.	21
B.	35 U.S.C. § 325(d).....	22
IX.	IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE....	22
A.	Statutory Grounds for Challenges	22

B.	Ground 1: Claims 1-7, 15-22, and 25-27 are Obvious under 35 U.S.C. § 103 over PayPal	24
1.	Reasons to Combine	24
2.	Summary of Paypal Express Checkout.....	27
3.	Claim 1	28
4.	Claim 2.....	72
5.	Claim 3	75
6.	Claim 4.....	75
7.	Claim 5.....	76
8.	Claim 6.....	77
9.	Claim 7	83
10.	Claim 15	84
11.	Claim 16.....	89
12.	Claim 17	89
13.	Claim 18	90
14.	Claim 19	90
15.	Claim 20	90
16.	Claim 21	90
17.	Claim 22	92
18.	Claim 25	92
19.	Claim 26	93

20.	Claim 27	94
C.	Ground 2: Claims 8-14, and 23-24 are Obvious Under 35 U.S.C. § 103 over Paypal and Schlesser.	94
1.	Summary of Schlesser.....	94
1.	Reasons to Combine Paypal and Schlesser.....	94
2.	Claim 8.....	97
3.	Claim 9.....	99
4.	Claim 10.....	100
5.	Claim 11	100
6.	Claim 12.....	100
7.	Claim 13	100
8.	Claim 14.....	100
9.	Claim 23	100
10.	Claim 24.....	103
X.	CONCLUSION.....	103
XI.	MANDATORY NOTICES	104
A.	Real Party-in-Interest	104
B.	Related Matters.....	104
C.	Lead and Back-up Counsel and Service Information	105
XII.	CLAIM APPENDIX.....	106

Petitioner's Exhibit List

Ex.1001	U.S. Patent No. 11,620,621
Ex.1002	Prosecution History of U.S. Patent No. 11,620,621 to Pollin et al.
Ex.1003	Declaration of Stephen C. Mott under 37 C.F.R. § 1.68
Ex.1004	<i>Curriculum Vitale</i> of Stephen C. Mott
Ex.1005	U.S. Provisional Application No. 61/655,482 to Pollin et al.
Ex.1006	PayPal Express Checkout Integration Guide
Ex.1007	SOAP API Developer Reference
Ex.1008	Sandbox User Guide
Ex.1009	Preparing for the New Updated Checkout Experience
Ex.1010	Express Checkout User Interface Standards
Ex.1011	PayPal Privacy Policy
Ex.1012	Digital Goods Integration Guide
Ex.1013	U.S. Patent Publication No. 2010/0094755 to Kloster
Ex.1014	HTML & Web Design
Ex.1015	U.S. Patent No. 9,830,596 to Collison et al.
Ex.1016	U.S. Patent Publication No. 2010/0327054 to Hammand
Ex.1017	U.S. Patent Publication No. 2012/0303516 to Fellig et al.
Ex.1018	U.S. Patent Publication No. 2007/0094100 to Hodson et al.
Ex.1019	Dynamic URLs vs. static URLs, Google Search Central Blog, Google for Developers
Ex.1020	U.S. Patent Publication No. 2012/0197807 to Schlesser <i>et al.</i>

Ex.1021	PayPal Technical Documentation
Ex.1022	PayPal Home Page
Ex.1023	Instant Payment Notification Guide
Ex.1024	Name-Value Pair API Developer Guide
Ex.1025	PayPal Merchant Setup and Administration Guide

I. INTRODUCTION

U.S. Patent No. 11,620,621 (the “’621 patent,” Ex.1001) claims well-known concepts for processing a payment between a payer and payee. The ’621 patent claims a technique by which a buyer is able to complete payment on a merchant’s website such that the merchant never receives the buyer’s sensitive financial data, such as a credit card number. This technique uses a third-party server that receives the buyer’s sensitive financial data and then provides the merchant with payment. The merchant can thus get paid without ever receiving the buyer’s financial data.

This Petition demonstrates that such payment techniques were already known. For example, PayPal provided a service that allowed merchants to receive payment without ever receiving the buyer’s sensitive financial account information. As described herein and supported by Dr. Mott’s declaration (Ex.1003), the PayPal Express Checkout documentation renders the challenged claims obvious.

Accordingly, pursuant to 35 U.S.C. §§ 311, 314(a), and 37 C.F.R. § 42.100, M&A Ventures, LLC, Repay Holdings Corporation, and Repay Holdings, LLC (“Repay” or “Petitioner”) respectfully requests that the Board review and cancel as unpatentable under 35 U.S.C. §103 claims 1-27 (hereinafter, the “Challenged Claims”) of the ’621 patent.

II. GROUNDS FOR STANDING

Petitioner certifies that the '621 patent is eligible for IPR and that Petitioner is not barred or estopped from requesting IPR challenging the patent claims. 37 C.F.R. § 42.104(a).

III. NOTE

Petitioner cites to exhibits' original page numbers. **Emphasis** in quoted material has been added. Claim terms are presented in *italics*.

IV. SUMMARY OF THE '621 PATENT

The '621 patent relates to “systems and methods for obtaining and using account information to process financial payments.” Ex.1001, 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.” Ex.1001, 7:8-12; *see also* 5:62-6:4. A “[s]ecure server 202...stores the financial account information received from the customer.” Ex.1001, 7:31-34; Ex.1003, ¶¶38-41.

The merchant server initiates two separate and distinct functions during the payment process: the financial account registration function and the token retrieval function. Ex.1001, 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. Ex.1003, ¶¶43-44. Through that communication link, the payer can submit her sensitive financial account information to the secure server to be stored and tokenized. Ex.1001, 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. Ex.1001, 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.” Ex.1001, 9:55-59; Ex.1003, ¶¶42-44.

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).” Ex.1001, 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.” Ex.1001, 8:40-43; Ex.1003, ¶45.

Third, to process the payment from the payer, the merchant then initiates the token retrieval function. Ex.1003, ¶46. 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.” Ex.1001, 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.” Ex.1001, 9:12-16. “The operator of the secure server 202 then processes the payment using the financial account pointed to by the token.” Ex.1001, 9:16-19; Ex.1003, ¶¶37-46.

V. LEVEL OF ORDINARY SKILL IN THE ART

A Person of Ordinary Skill in The Art (“POSITA”) on or before June 5, 2012 would have been knowledgeable and familiar with the field of payment processing systems. That person would have a bachelor’s degree related to the field of payment processing systems and approximately two years of experience working in that field. Lack of education can be remedied by additional work experience, and vice versa. Ex.1003, ¶¶28-30.

VI. CLAIM CONSTRUCTION

In an *inter partes* review, claims “shall be construed using the same claim construction standard that would be used to construe the claim in a civil action under 35 U.S.C. 282(b), including construing the claim in accordance with the

ordinary and customary meaning of such claim as understood by one of ordinary skill in the art and the prosecution history pertaining to the patent.” 37 C.F.R. § 42.100(b). The Board only construes the claims to the extent necessary to resolve the underlying controversy. *Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017). Petitioner submits that for the purposes of this proceeding, other than the claim terms set forth below, the terms of the challenged claims should be given their plain and ordinary meaning.¹ Ex.1003, ¶51.

A. “*providing a non-sensitive electronic data token representing the sensitive financial account information to the merchant... without providing the non-sensitive electronic data token to the payer*”

This limitation should be construed as “*during execution of the token retrieval function, providing a non-sensitive electronic data token representing the sensitive financial account information to the merchant ... without providing the non-sensitive electronic data token to the payer.*”² Ex.1003, ¶52. In short, the claim

¹ Petitioner does not concede that any term in the challenged claims meets the statutory requirements of 35 U.S.C. § 112, or that the challenged claims recite patentable subject matter under 35 U.S.C. § 101.

² This limitation appears in claim 1 ([1.4.1]), claim 8 ([8.2.1.8.1]), and claim 25 ([25.4.1]). The same analysis for claim 1 applies to claims 8 and 25.

language only requires that, *during the token retrieval function*, the secure server does not provide the token to the payer when it provides the token to the merchant. It does not mean that the secure server must never provide the token to the payer. Ex.1003, ¶52.

Claim 1 includes two discrete functions: (1) the financial account registration function and (2) the token retrieval function. Ex.1003, ¶53. Claim 1 recites “*executes the financial account registration function, upon initiation by the merchant server, by*” ([1.1.5]) performing several steps recited in [1.1.5.1]-[1.1.5.4]. Claim 1 separately recites that “*executing a token retrieval function, upon initiation by the merchant server via the APP*” ([1.4]) includes the step of “*providing a non-sensitive electronic data token ... to the merchant...without providing the non-sensitive electronic data token to the payer...*” ([1.4.1]). Thus, as specifically delineated in the claim, providing the token “*without providing the non-sensitive electronic data token to the payer*” is performed during execution of the token retrieval function. Ex.1003, ¶54.

The prosecution history of the '621 patent supports this understanding. Ex.1003, ¶¶55-56. In order to overcome § 101 rejections, PO made amendments reciting separate financial account registration and token retrieval functions. These amendments grouped certain steps as being a part of one of the functions. *See* Ex.1002, 138-139. Thus, PO made clear that the limitations recited under the token

retrieval function, like “*providing a non-sensitive electronic data token to the merchant... without providing the non-sensitive electronic data token to the payer,*” are separate from the financial account registration function. *See Schindler Elevator Corp. v. Otis Elevator Co.*, 593 F.3d 1275, 1285 (Fed. Cir. 2010) (“[A]n amendment that clearly narrows the scope of a claim, such as by the addition of a new claim limitation, constitutes a disclaimer of any claim interpretation that would effectively eliminate the limitation or that would otherwise recapture the claim's original scope.”). Thus, the limitations recited with respect to the token retrieval function define what does and does not happen *during execution of the token retrieval function*. Ex.1003, ¶¶55-56.

Moreover, the specification describes the financial account registration and token retrieval functions as being distinct, stating that “[a] hosted system on a secure server 202 provides **Financial Account Registration functions and Token Retrieval functions.**” Ex.1001, 5:58-62; *see also* 6:63-7:1, 7:1-8. The specification describes the secure server sending the token to the merchant server during the token retrieval function but does not describe sending the token during any other step in the payment process. *See e.g.* Ex.1001, 7:31-34 (“Secure server 202...stores the financial account information received from the customer...and provides a token to merchant server 204.”). Notably, the specification never explicitly states that (1) the token is sent to the merchant server *without* being sent

to the payer during the token retrieval function or (2) the token *cannot* be sent to the payer during any other step of the payment process. Ex.1003, ¶58. In fact, the '621 patent states that the token is “only meaningful to the merchant,” implying that it is not meaningful to the payer. Ex.1001, 10:46-48; Ex.1003, ¶59. Thus, there would be no reason to prevent the token from being sent to the payer. Ex.1003, ¶¶57-59.

Additionally, the '482 application (incorporated by reference in the '621 patent) contemplates that the non-sensitive electronic data token *is* provided to the payer outside of execution of the token retrieval function. Ex.1003, ¶60. The '482 application explains that “[t]he hosted system [secure server] returns a token or other account identifier **to the user** [payer] and/or the merchant's system **when the financial account information has been registered at the hosted system.**” Ex.1005, 21; *see* 31-32 (the end user (payer) requests a reference ID from the secure server). Thus, the '482 application discloses that the token may be sent to the payer at other times such as during the financial account registration function. Ex.1003, ¶60.

Therefore, in view of the structure of the claims, the prosecution history, and the disclosure of the '621 patent, a POSITA would have understood that [1.4.1] requires the token be provided to the merchant without providing the token to the payer only during execution of the token retrieval function, and that the token *may*

be sent to the payer during another step in the payment process. Ex.1003, ¶¶52-61.

B. “receiving the non-sensitive electronic data token from the secure server, wherein...the secure server does not provide the non-sensitive electronic data token to the payer”

Similar to [1.4.1] described in Section VI.A, this limitation should be construed as “*during execution of the token retrieval function, receiving the non-sensitive electronic data token from the secure server, wherein...the secure server does not provide the non-sensitive electronic data token to the payer.*”³ Ex.1003, ¶62.

As with claim 1, claim 15 clearly divides the payment process into two discrete functions: (1) the financial account registration function and (2) the token retrieval function, and specifies that “*the secure server does not provide the non-sensitive electronic data token to the payer*” applies to the “*token retrieval function.*” Ex.1003, ¶¶63-65.

Further, during the “*financial account registration function*” ([15.6]), claim 15 recites the encrypted financial account information is transmitted to the secure server “*without providing the sensitive financial account information to the merchant server*” ([15.6.3]). Similarly, during the “*token retrieval function*”

³ This limitation appears in claim 15 ([15.8]) and claim 23 ([23.2.8]). The same analysis for claim 15 applies to claim 23.

([15.7]), claim 15 also recites the merchant server receives the non-sensitive electronic data token “*wherein the merchant server does not receive the sensitive financial account information*” ([15.8]). Therefore, claim 15 specifies that the financial account information is not sent to the merchant server **both** during the financial account registration function **and** during the token retrieval function. Ex.1003, ¶66. In contrast, the claim only requires “*the secure server does not provide the non-sensitive electronic data token to the payer*” during the token retrieval function. Ex.1003, ¶66.

As with claim 1, during prosecution, PO amended pending claim 31 (issued claim 15) to recite separate financial account registration and token retrieval functions and grouping certain steps as being a part of one of the functions. *See* Ex.1002, 125-126. By making amendments reciting a separate financial account registration function and token retrieval function, PO made clear that the limitations under the token retrieval function, like “*the secure server does not provide the non-sensitive electronic data token to the payer*” ([15.8]), are separate from the financial account registration function. Ex.1003, ¶¶67-68; *see Schindler Elevator*, 593 F.3d at 1285.

For the same reasons discussed in Section VI.A, the specification supports this understanding of [15.8]. Ex.1003, ¶69.

Therefore, in view of the structure of the claims, the prosecution history, and the disclosure of the '621 patent, a POSITA would have understood that [15.8] requires the secure server to not provide the token to the payer only during the token retrieval function and that the token may be sent to the payer during another step in the payment process. Ex.1003, ¶¶62-70.

C. “*generating a uniform resource locator (URL), ... the URL comprising either: a dynamic URL generated by the secure server...; or a static URL and a hypertext transport protocol (HTTP) parameter...*”

The claims recite that the step of “*generating*” a URL can be done either dynamically or statically. Ex.1003, ¶71. The generating step should be construed such that “*generating*” the URL statically includes the secure server “*generating*” the URL by prompting the merchant server to combine a predefined static URL with an “*HTTP parameter.*” Ex.1003, ¶71. In other words, “*generating a uniform resource locator*” does not strictly require that the URL be generated entirely at the secure server. Ex.1003, ¶71.

The claim language itself informs this construction. Ex.1003, ¶72. Claim 1 recites that “*generating a URL*” ([1.1.5.1]) includes either: (1) “*a dynamic URL generated by the secure server*” ([1.1.5.1a]) or (2) “*a static URL*” known to the merchant and a “*hypertext transport protocol (HTTP) parameter used by the*

secure server to identify the payer and the payee” ([1.1.5.1b]).⁴ For (1) the dynamic URL, claim 1 explicitly recites that the dynamic URL be “*generated by the secure server*” ([1.1.5.1a]). Conversely, for (2) the static URL, the claim notably does not recite that the static URL be generated “*by the secure server.*” Accordingly, the claim language for the static URL is drafted broadly so as to include the example where “generating” the URL is done by prompting the merchant server to combine a static URL with an HTTP parameter. Ex.1003, ¶72.

Indeed, the specification of ’621 patent distinguishes between the dynamic method (in which the URL is generated by the secure server), and the static method (in which the URL is generated by prompting the merchant server to combine a static URL with the HTTP parameter). Ex.1003, ¶¶73-85. During the dynamic URL method (1A), the secure server executes subroutine 3. Ex.1001, 11:60-62. Subroutine 3 provides “a financial account registration entry form.” Ex.1001, 11:60-62. “In step 430 [of subroutine 3], the API [of the secure server] **creates a registration URL.**” Ex.1001, 14:46-47. Thus, the “registration URL” (dynamic URL) is generated in subroutine 3. Ex.1003, ¶76.

⁴ This limitation also appears in claim 8 ([8.2.1.5.1]-[8.2.1.5.1b]). The same analysis for claim 1 applies to claim 8.

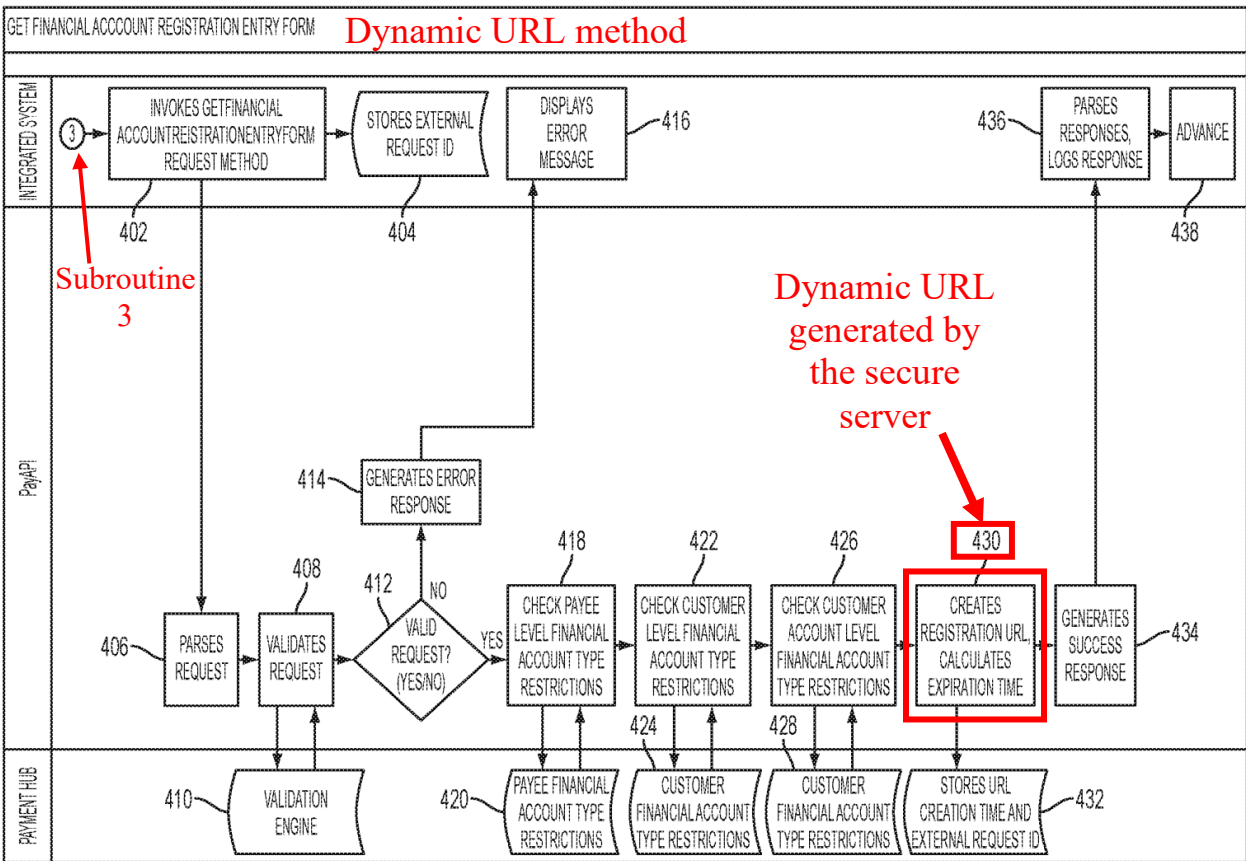


FIG. 4g

Ex.1001, Fig. 4g (annotated); Ex.1003, ¶75

Contrastingly, when static URL method (1B) is used, subroutine 3 is **not** performed and the secure server does **not** generate a new URL. Ex.1003, ¶¶79-83. Thus, step 430 of subroutine 3 in which the secure server “generates” the registration URL is **not** performed. Ex.1003, ¶81. Instead, an SSL connection is established using a static URL combined with an array of codes: “The payment application then displays **an array of codes** in step 548 and launches a web browser with **a financial account registration URL** [in step] 550” (Ex.1001,

12:58-60):

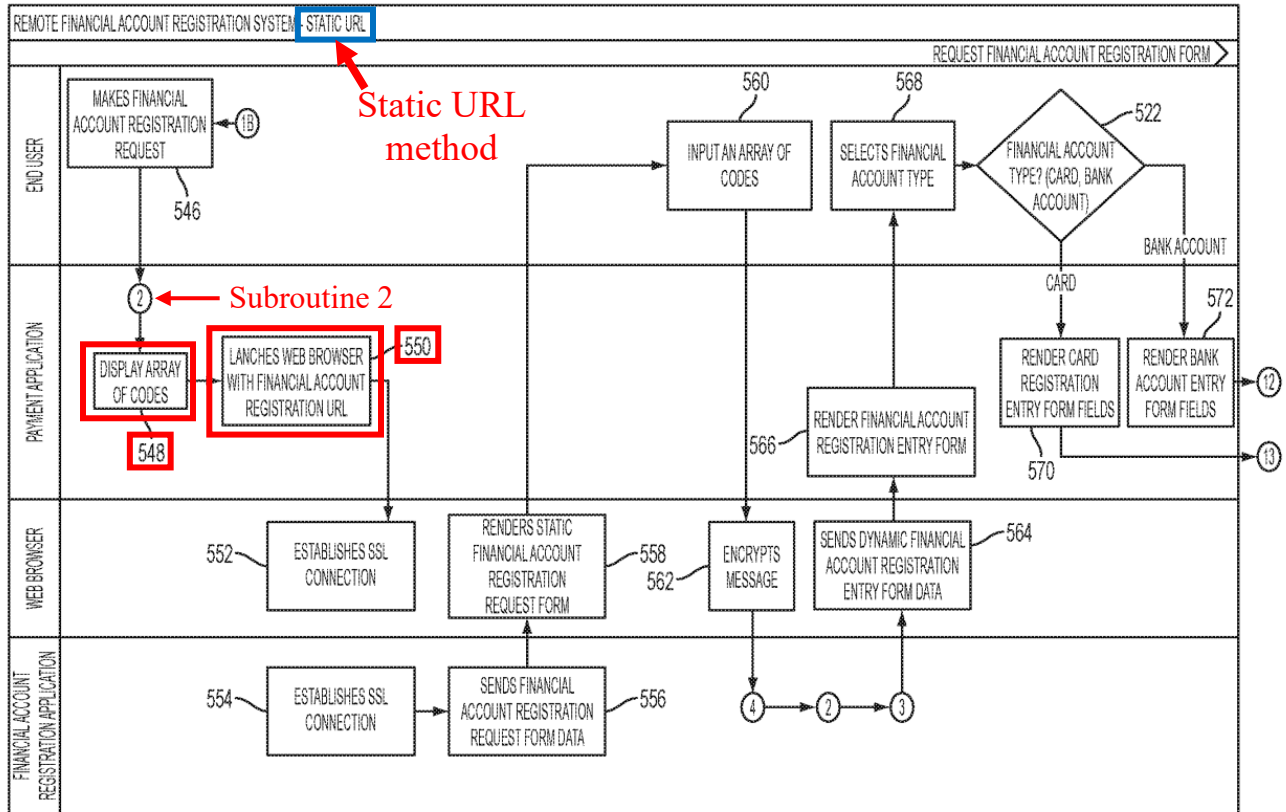


FIG. 4d

Ex.1001, Fig. 4d (annotated); Ex.1003, ¶82

The static URL method (1B) thus never describes the secure server *generating* a new financial account registration URL. Ex.1003, ¶83. Rather, a POSITA would have understood that the payment application (merchant server) uses a predefined static URL and the array of codes (parameters) to establish the SSL connection. Ex.1003, ¶83; *see* Ex.1001, 11:28-34.

Additionally, the '482 application (incorporated by reference in the '621

patent) similarly distinguishes how the dynamic and static URLs are generated. Ex.1003, ¶84. The '482 application explains that, in the dynamic URL method, a “temporary URL” is requested from PayAPI (secure server). Ex.1005, 62. In the static URL method, however, “a temporary array of codes” is requested from PayAPI and submitted on a Financial Account Registration site. Ex.1005, 62. Thus, the Financial Account Registration site (static URL) is not requested from and, thus, not generated anew by the secure server. Ex.1003, ¶85.

For these reasons, a POSITA would have understood that, when the API “*generat[es] a... URL*” comprising “*a static URL*,” the static URL itself is “generated” by causing a static URL to be combined with a “*parameter*” “*for establishing the secure socket layer connection.*” Ex.1003, ¶¶71-86.

VII. RELIEF REQUESTED

Petitioner asks that the Board institute a trial for *inter partes* review and cancel the Challenged Claims in view of the analysis below.

VIII. DISCRETIONARY DENIAL WOULD BE INAPPROPRIATE

A. The Fintiv Factors.

The Fintiv factors favor institution. (1) No motion to stay has been filed. (2) The trial date is irrelevant in light of Petitioner’s stipulation below. (3) The copending litigations are in relatively early stages. (4) Petitioner hereby stipulates that if instituted, Petitioner will not pursue in District Court the specific grounds

asserted here, or on any other ground that was raised or could have been reasonably raised in this IPR. For this reason alone, the Board should not discretionarily deny institution in view of the parallel proceedings. *See* June 21, 2022 Director Memo, p. 3. (5) Petitioner’s involvement in the parallel proceeding should not be a basis for denying institution. (6) The merits of this case are compelling. Accordingly, discretionary denial is inappropriate here.

B. 35 U.S.C. § 325(d)

Denial under § 325(d) is not warranted because the challenges presented in this petition are neither cumulative nor redundant to the prosecution of the ’621 patent.

IX. IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE

A. Statutory Grounds for Challenges

This petition challenges the ’621 patent as being obvious in view of the teachings of the identified prior art references as well as the knowledge of a POSITA.

Grounds	Claim(s)	Basis
#1	1-7, 15-22, and 25-27	35 U.S.C. § 103 in view of PayPal
#2	8-14 and 23- 24	35 U.S.C. § 103 in view of PayPal and Schlesser

PayPal includes ten documents, as described in Section IX.B.1.

PayPal Express Checkout Integration Guide (Ex.1006) was initially published on June 30, 2008 and most recently revised on January 24, 2011, and thus is prior art under at least pre-AIA 35 U.S.C. § 102(b).

PayPal SOAP API Developer Reference (Ex.1007) was initially published in October 2006 and most recently revised on April 5, 2011, and thus is prior art under at least pre-AIA 35 U.S.C. § 102(b).

PayPal Sandbox User Guide (Ex.1008) was initially revised in December 2005 and most recently revised in July 2010, and thus is prior art under at least pre-AIA 35 U.S.C. § 102(b).

PayPal Preparing for the New Updated Checkout Experience (Ex.1009) was initially published in April 2010 and most recently revised in May 2012, and thus is prior art under at least pre-AIA 35 U.S.C. § 102(a).

PayPal Express Checkout User Interface Standards (Ex.1010) was most recently updated in October 2009, and thus is prior art under at least pre-AIA 35 U.S.C. § 102(b).

PayPal Privacy Policy (Ex.1011) was most recently updated on July 13, 2010, and thus is prior art under at least pre-AIA 35 U.S.C. § 102(b).

PayPal Digital Goods Integration Guide – EC Edition (Ex.1012) was initially published on January 26, 2011 and most recently revised on January 30, 2012, and thus is prior art under at least pre-AIA 35 U.S.C. § 102(a).

PayPal Instant Payment Notification Guide (Ex.1023) was initially published on February 16, 2009 and most recently revised on July 17, 2010, and thus is prior art under at least pre-AIA 35 U.S.C. § 102(a).

PayPal Name-Value Pair API Developer Guide (Ex.1024) was initially published in October 2006 and most recently revised on April 5, 2011, and thus is prior art under at least pre-AIA 35 U.S.C. § 102(a).

PayPal Merchant Setup and Administration Guide (Ex.1025) was initially filed in June 2009 and most recently revised in January 2010, and thus is prior art under at least pre-AIA 35 U.S.C. § 102(a).

U.S. Published Application US2012/0197807 to Schlessner et al. (“**Schlessner**” Ex.1020) was filed January 28, 2011 and published August 2, 2012. Schlessner is prior art under at least pre-AIA 35 U.S.C. § 102(a).

B. Ground 1: Claims 1-7, 15-22, and 25-27 are Obvious under 35 U.S.C. § 103 over PayPal

1. Reasons to Combine

The PayPal reference includes a series of documents that describe the PayPal Express Checkout Product. PayPal Express Checkout (“PayPal”) is a product by PayPal that allows a buyer to make purchases on a merchant’s website using PayPal without the merchant receiving sensitive financial information from the

buyer. PayPal is described in ten printed publications that were publicly available on PayPal's website, including:

- PayPal Express Checkout Integration Guide (Ex.1006),
- PayPal SOAP API Developer Reference (Ex.1007),
- PayPal Sandbox User Guide (Ex.1008),
- PayPal Preparing for the New Updated Checkout Experience (Ex.1009),
- PayPal Express Checkout User Interface Standards (Ex.1010),
- PayPal Privacy Policy (Ex.1011),
- PayPal Digital Goods Integration Guide (Ex.1012),
- PayPal Instant Payment Notification Guide – EC Edition (Ex.1023),
- PayPal Name-Value Pair API Developer Guide (Ex.1024), and
- PayPal Merchant Setup and Administration Guide (Ex.1025).

Although these are separate documents, they each cover the same product and, thus, are considered a single prior art reference herein (collectively as “PayPal”).⁵ Ex.1003, ¶90. All of these documents (aside from the Privacy Policy

⁵ Note, PayPal is directed to merchants and uses second-person perspective (“you” and “your”). Herein, the second-person language in the quotes is replaced with “the merchant” or similar for clarity.

and Merchant Setup and Administration Guide) were linked under “Express Checkout” on PayPal’s Technical Documentation page (Ex.1021) before the priority date of the ’621 patent.⁶ Ex.1003, ¶91. The Merchant Setup and Administration Guide was linked under “Merchant Setup and Administration Guide” on the Technical Documentation page just below the other “Express Checkout” documents. Ex.1003, ¶92. The Privacy Policy was easily accessible on PayPal’s website and can be accessed via two clicks from the Technical Documentation page. Ex.1003, ¶¶93-94. Thus, a POSITA would have understood that because each of these documents describes aspects of the same product—PayPal’s Express Checkout—the teachings from each of the documents act as a single reference. Ex.1003, ¶95. In other words, these documents together are akin to a series of chapters in a single book describing PayPal’s Express Checkout product.

Further, a POSITA would have found it obvious to combine the teachings of

⁶ Exhibits 1021 and 1022 are documents from the Internet archive, showing that as of at least July 10, 2011 (Ex.1021), the PayPal Express Checkout documentation relied on herein was publicly available. A POSITA looking to implement PayPal’s technique would have known to look at PayPal’s website under the technical documentation section. Ex.1003, ¶¶91-95.

each separate document—in fact, it would have been necessary to combine all of the teachings. Ex.1003, ¶96. Each of the documents are the equivalent of a chapter in a single manual describing PayPal’s Express Checkout product. Ex.1003, ¶96. To gain a full and complete understanding of PayPal’s Express Checkout product, a POSITA would have necessarily combined these documents, because only by combining the documents could a POSITA implement the operation and functionality of an existing product on the market. Ex.1003, ¶96. In other words, the combination of these documents is not hypothetical—POSITAs were expected to combine the teachings in order to implement PayPal’s Express Checkout product. Ex.1003, ¶96. It would have been desirable and beneficial to implement PayPal’s product in the tried-and-true method provided by the producer of that product. Ex.1003, ¶96. Accordingly, combining the teachings of the individual documents is merely a matter of combining prior art elements (the teachings of each individual document) according to known methods to yield predictable results. Ex.1003, ¶96; *see also* ¶¶97-105.

2. Summary of PayPal

PayPal describes the same tokenization process that is claimed in the ’621 patent. Like the ’621 patent, PayPal “enables [the merchant] to accept PayPal while retaining control of the buyer and the overall checkout flow.” Ex.1006, 9. A more detailed summary of PayPal can be found in Dr. Mott’s Declaration.

Ex.1003, ¶¶106-112.

3. Claim 1

[1.0] 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:

First, PayPal describes a method for processing a checkout transaction (“[a] method of processing a payment transaction from a payer to a payee”). Ex.1003, ¶¶113-116. During a transaction, the buyer “[c]hooses Express Checkout,” “[r]eviews the transaction on PayPal,” and “[c]onfirms the order and pays from [the merchant’s] site” (Ex.1006, 10):



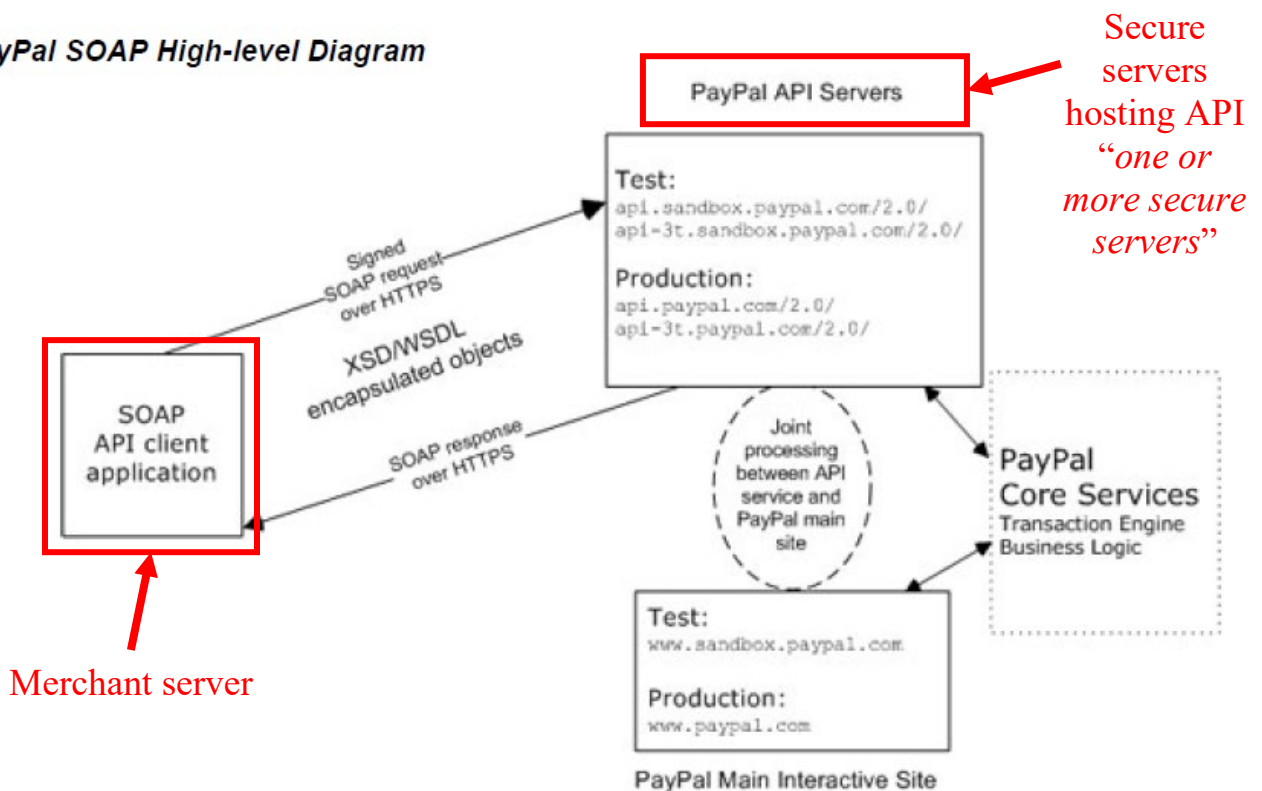
Ex.1006, 10; Ex.1003, ¶113

When the buyer makes a purchase on the merchant’s website, the merchant uses API operations to “set[] up the transaction, obtain[] information about the buyer, and handle[] the payment and complete[] the transaction.” Ex.1006, 17. These API operations include DoExpressCheckoutPayment, which “[c]ompletes the Express Checkout transaction.” Ex.1006, 17; *see also* 21-23 (“To complete an

Express Checkout transaction, [the merchant] must invoke the DoExpressCheckoutPayment API operation.”), 26-27.

Second, PayPal utilizes a client-server model and is thus “*performed by one or more secure servers.*” Ex.1003, ¶¶117-119. “The PayPal API uses a client-server model in which [the merchant’s] website is a client of the PayPal server.” Ex.1006, 61. “The PayPal SOAP API provides programmatic access to PayPal features and services” to the merchant. Ex.1007, 11. “PayPal SOAP is a combination of client-side and server-side schemas, hardware and software servers, and core services” (Ex.1007, 11):

PayPal SOAP High-level Diagram



Ex.1007, 11 (annotated); Ex.1003, ¶117

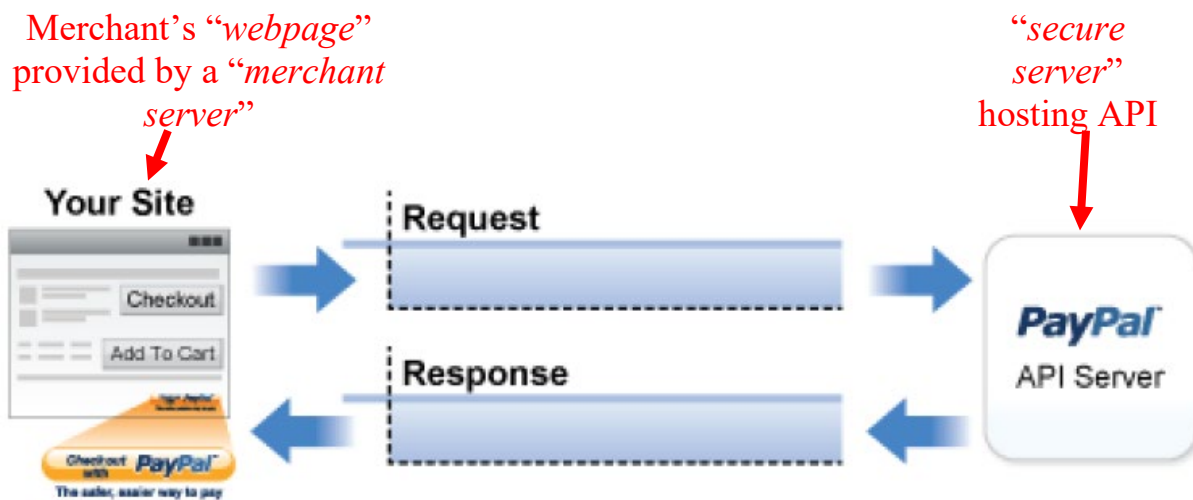
The merchant processes the transaction from the buyer using several API operations (e.g., DoExpressCheckoutPayment) to instruct the PayPal API servers to mediate the transaction. Ex.1006, 62 (“A PayPal API server performs the operation and returns a response.”).

PayPal’s servers are “*secure*” because “[t]he PayPal SOAP API service is protected to ensure that only authorized PayPal members use it” and uses “four levels of security” including a “required API username...and API password,” “a third authentication mechanism” that is either an API Certificate or Signature, and “Secure Socket Layer (SSL) data transport.” Ex.1007, 13. “A failure of authenticated security at any one of these levels denies access to the PayPal SOAP API service.” Ex.1007, 13. Because communication to PayPal’s servers via PayPal SOAP API is protected, a POSITA would have understood that PayPal’s servers are “*secure servers*.” Ex.1003, ¶119. Moreover, PayPal refers to PayPal’s servers as “secure” throughout. See e.g. Ex.1006, 13 (“secure PayPal servers”), Ex.1010, 11 (“PayPal’s secure servers”). Ex.1003, ¶¶113-120.

[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:

First, PayPal’s servers act as an API server (“*provid[e] ... an application programming interface (API)*”). Ex.1003, ¶¶121-122. “The PayPal SOAP API

provides programmatic access to PayPal features and services.” Ex.1007, 11. The PayPal SOAP API allows the merchant to communicate with PayPal’s servers. *See* 61 (“The PayPal API uses a client-server model in which [the merchant’s] website is a client of the PayPal server. A page on [the merchant’s] website initiates an action on a PayPal API server by sending a request to the server. The PayPal server responds with a confirmation that the requested action was taken.”). The diagram below illustrates communication between the merchant server and secure server:



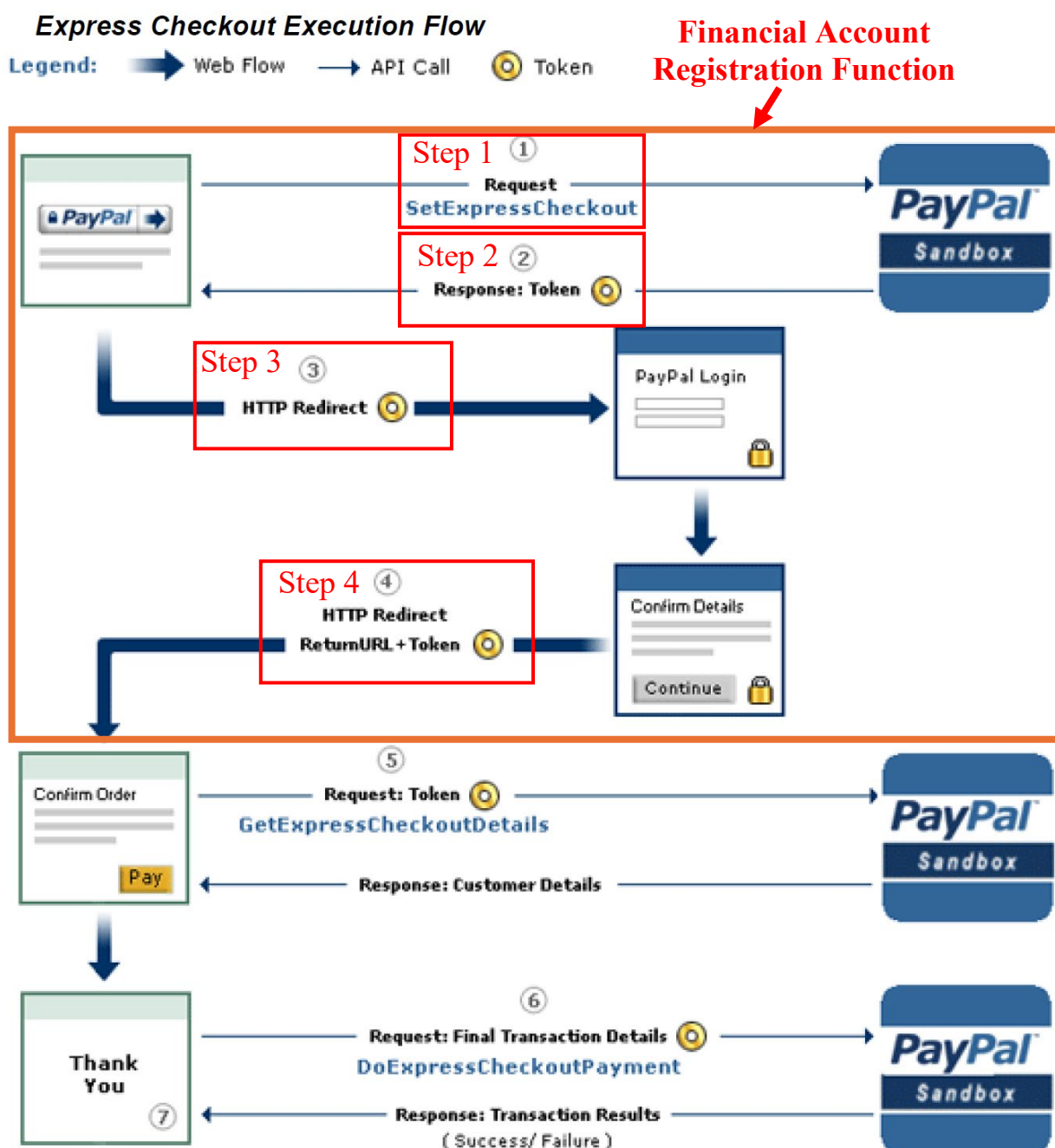
Ex.1006, 61 (annotated); Ex.1003, ¶121

Second, the API is provided to “a merchant server providing a webpage to a payer computing system used by the payer.” Ex.1003, ¶¶123-127. PayPal discloses that the merchant “integrate[s] Express Checkout on [its] website” to “optimize [its] website’s checkout flow and significantly improve [the] buyer’s experience.” Ex.1006, 15; Ex.1010, 7. The merchant implements express checkout “by placing

the Checkout with PayPal button on [its] Shopping Cart page and by placing the PayPal mark on [its] Payment Methods page.” Ex.1006, 12. The buyer can access the merchant’s website via, for example, a personal computer. *See e.g.* Ex.1006, 55 (Express Checkout can “run[] on personal computers.”). The merchant’s website is hosted by a merchant’s server. *See* Ex.1007, 12-13; *see also* Ex.1006, 14, 42; Ex.1003, ¶¶121-128.

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

First, PayPal’s API provides a “*financial account registration*” function (Ex.1003, ¶¶132-135), which includes steps 1-4 of the Example Transaction shown below:



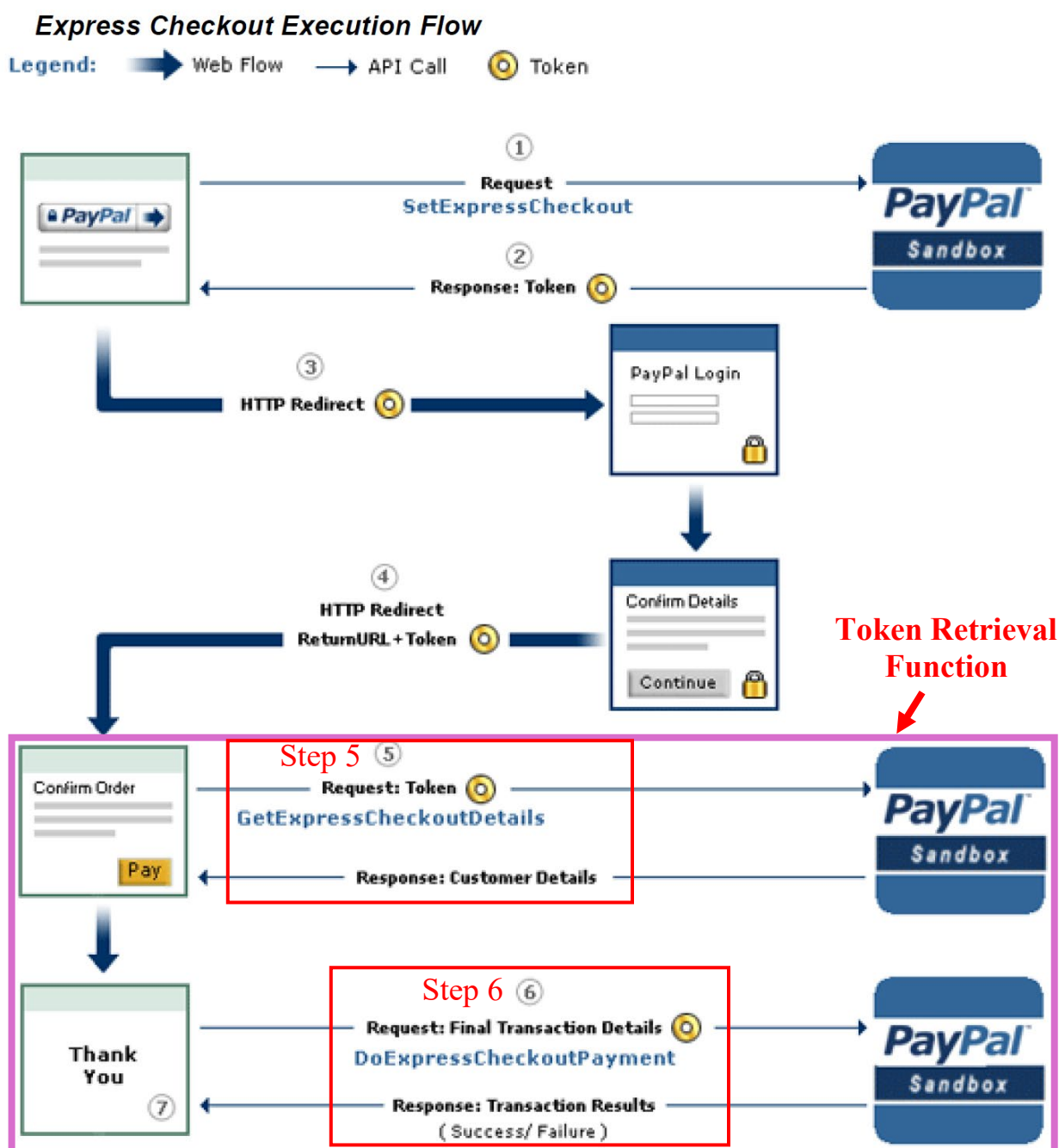
Ex.1006, 24 (annotated); Ex.1003, ¶132

When a buyer initiates a transaction on the merchant's website, the merchant sets up the Express Checkout transaction by “invok[ing] the SetExpressCheckout

API operation to provide sufficient information to initiate the payment flow and redirect to PayPal if the operation was successful.” Ex.1006, 19; *see also* 24-25 (Step 1). SetExpressCheckout returns a token. Ex.1006, 17, 21; *see* 25 (Step 2).

After setting up the transaction, the merchant redirects the buyer to PayPal’s website. “PayPal provides a command that [the merchant] use[s] when redirecting [the] buyer’s browser to PayPal.” Ex.1006, 17. “This command enables [the] buyer to log into PayPal to approve an Express Checkout payment.” Ex.1006, 17. The merchant “redirect[s] the buyer’s browser to PayPal and execute[s] the express-checkout command using the token returned in the SetExpressCheckout response.” Ex.1006, 21; *see also* 25 (Step 3). “If the buyer approves the payment, PayPal redirects to the success URL” the merchant specified in SetExpressCheckout. Ex.1006, 17; 25-26 (Step 4).

Second, PayPal’s API provides a “*token retrieval function[]*” (Ex.1003, ¶¶136-141), which includes steps 5-6 of the Example Transaction shown below:



Ex.1006, 24 (annotated); Ex.1003, ¶136

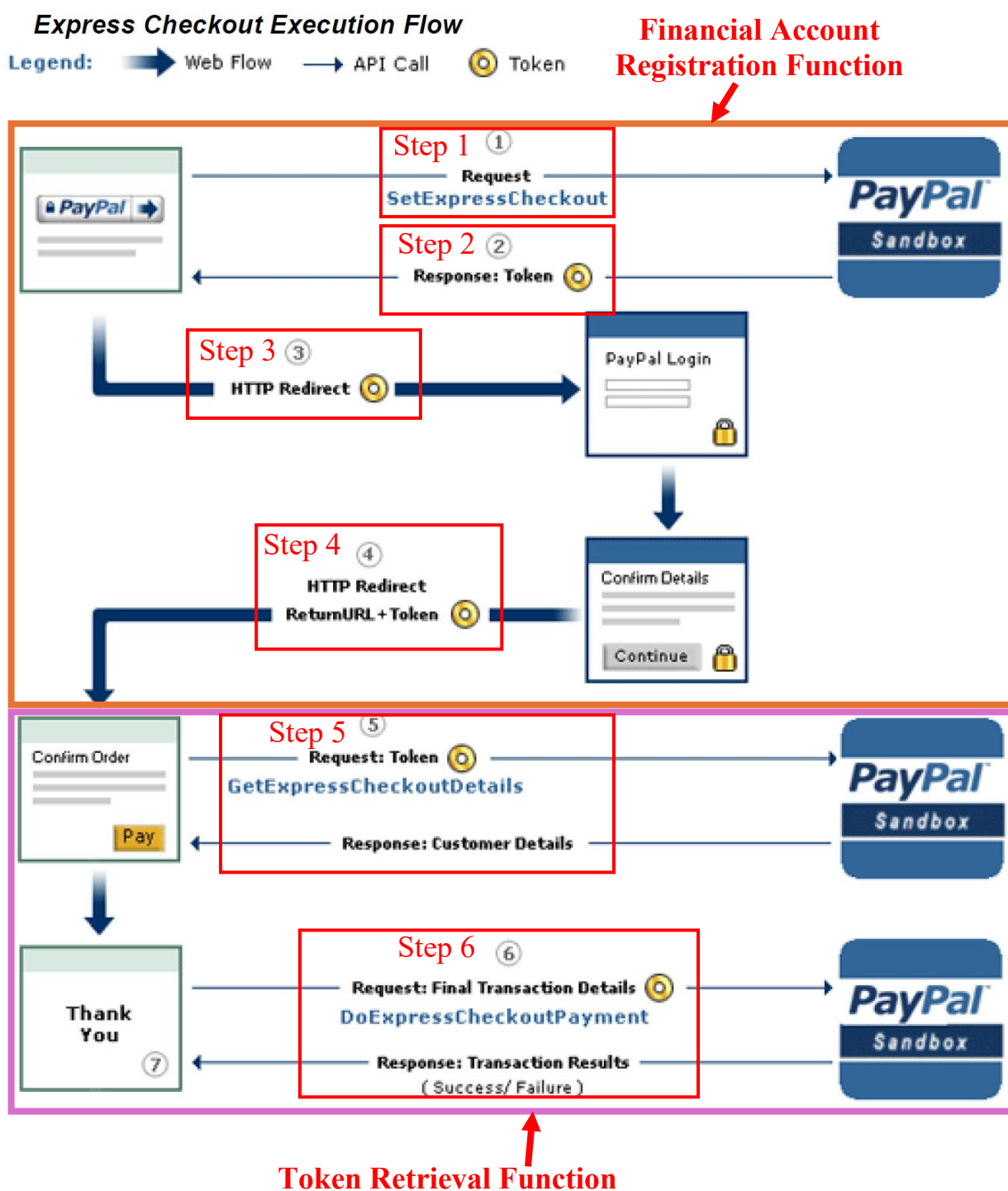
After the buyer “approves the payment” and “PayPal redirects to the success URL,” the merchant “invoke[s] the GetExpressCheckoutDetails API operation.”

Ex.1006, 17, 21; *see also* 26 (Step 5). The merchant “take[s] this action to obtain information about the buyer,” including the Payer ID. Ex.1006, 21; *see also* 26. The merchant calls GetExpressCheckoutDetails by “[s]pecify[ing] the token.” Ex.1006, 21.

After receiving the Payer ID, the merchant completes the transaction by “invok[ing] the DoExpressCheckoutPayment API operation.” Ex.1006, 21; *see* 21-22, 26-27 (Step 6). The merchant calls DoExpressCheckoutPayment by “[s]pecify[ing] the token returned by PayPal when it redirects the buyer’s browser back to [the merchant’s] site,” “[s]pecify[ing] the Payer ID,” “[s]pecify[ing] the amount of the order,” and “[s]pecify[ing] the payment action.” Ex.1006, 22; *see also* 26-27. The steps of the Example Transaction can thus “*be executed to process the payment transaction.*” Ex.1003, ¶¶129-142.

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

As discussed in [1.1.1], PayPal includes SetExpressCheckout and _ExpressCheckout functions (“*financial account registration*” function) and the GetExpressCheckoutDetails and DoExpressCheckoutPayment functions (“*token retrieval function[]*”):



Ex.1006, 24 (annotated); Ex.1003, ¶143

PayPal’s API provides merchant servers access to these functions. “PayPal

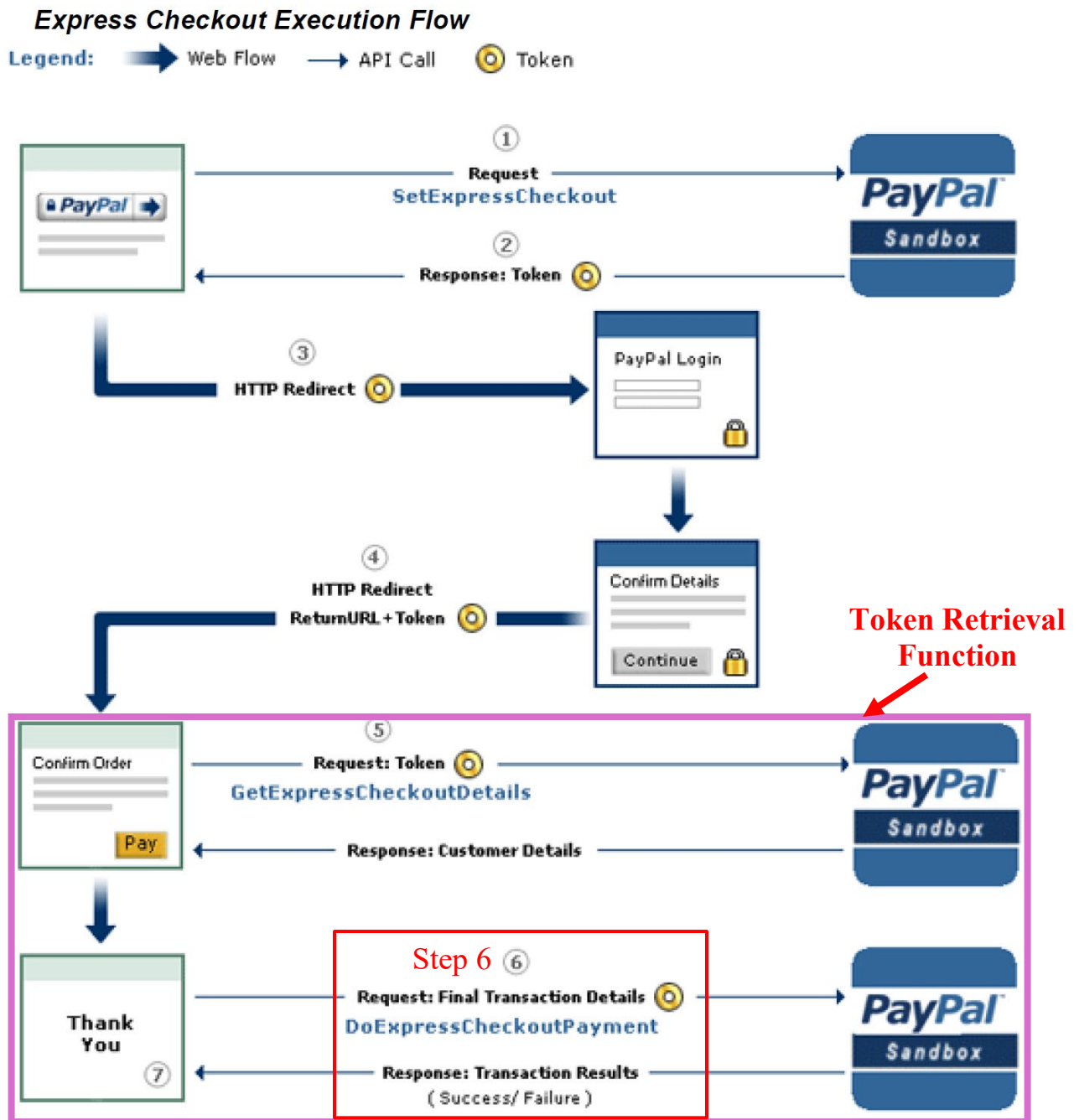
provides a command [_ExpressCheckout] that [the merchant] use[s] when redirecting [the] buyer's browser to PayPal." Ex.1006, 17. Moreover, "The PayPal API provides three API operations for Express Checkout," which include SetExpressCheckout, GetExpressCheckoutDetails, and DoExpressCheckoutPayment. Ex.1006, 17. "The PayPal SOAP API provides programmatic access to PayPal features and services" to the merchant, including the three express checkout API operations, which are used for both functions. Ex.1007, 11; Ex.1003, ¶¶143-146.

[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;

First, PayPal discloses a token ("*at least one data element*") that is "*associated with the payer.*" Ex.1003, ¶147. "Express Checkout uses a token to control access to PayPal and execute Express Checkout API operations." Ex.1006, 18. The "token is a value assigned by PayPal **that associates the execution of API operations and commands with a specific instance of a user experience flow.**" Ex.1006, 16. Thus, the token is a "*data element associated with the payer.*" Ex.1003, ¶147.

Second, PayPal's servers receive the token ("*data element*") and "*a payment amount from the payer to the payee.*" Ex.1003, ¶148. When invoking DoExpressCheckoutPayment, the merchant "[s]pecif[ies] **the token**" and

“[s]pecif[ies] **the amount of the order.**” Ex.1006, 22. Accordingly, the merchant sends the token (“*data element*”) and the amount of the order (“*payment amount*”) to PayPal’s servers, as illustrated in Step 6 of the Example Transaction:



Ex.1006, 24 (annotated); Ex.1003, ¶148

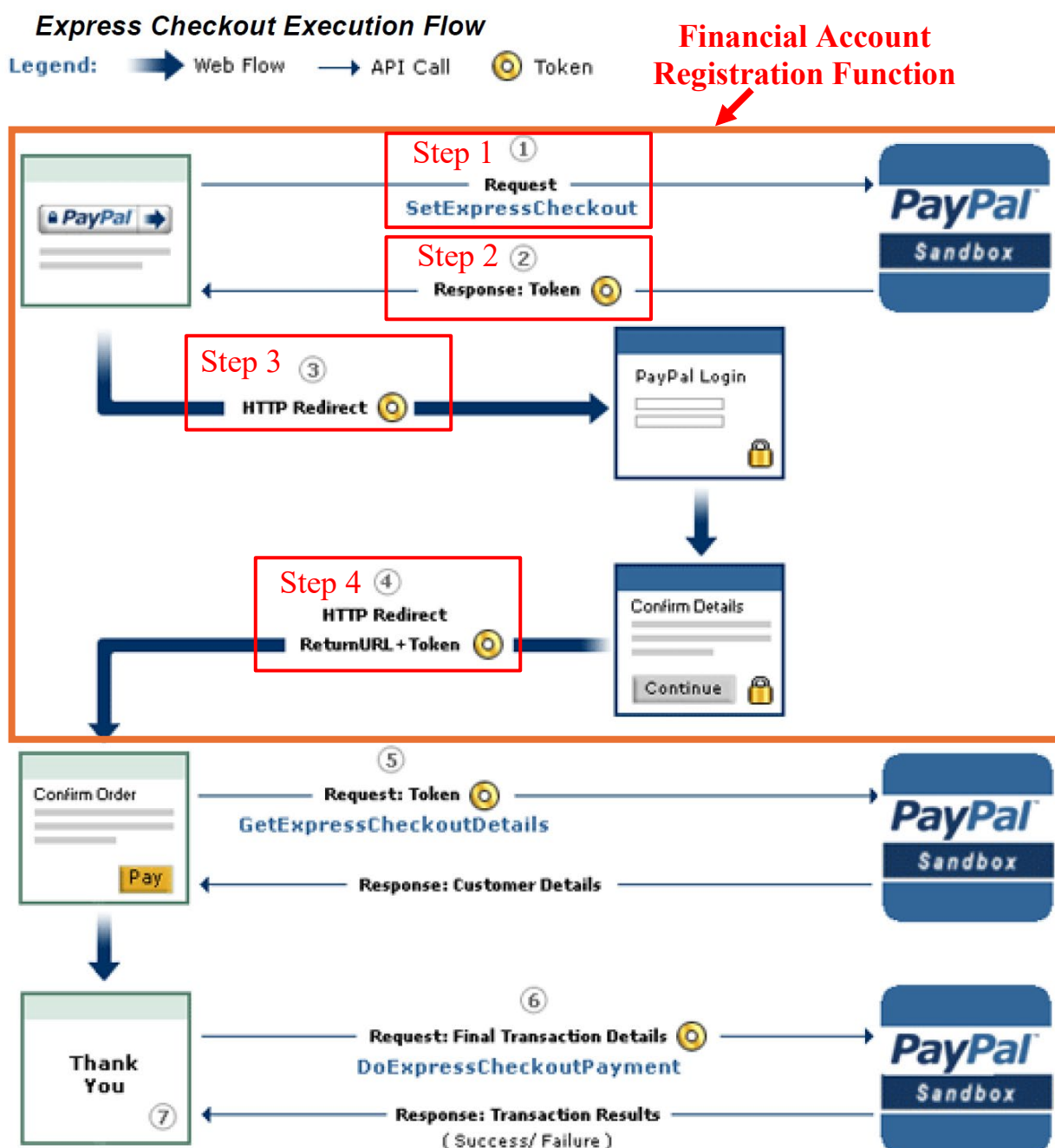
Thus, because PayPal discloses that the merchant specifies the token identifying the transaction and the amount of the order in the DoExpressCheckoutPayment request, PayPal renders this limitation obvious. Ex.1003, ¶¶147-149.

[1.1.4] *authenticates the payee; and*

PayPal “*authenticates the payee*” using various levels of security to prevent unauthorized access of PayPal’s servers, including “[a] required API username (Username field) and API password (Password field)” and “[a] third required authentication mechanism” including either an “API Certificate” or an “API Signature.” Ex.1007, 13. “A failure of authenticated security at any one of these levels denies access to the PayPal SOAP API service.” Ex.1007, 13. “To use the PayPal API, [the merchant] must have API credentials that identify [it] as a PayPal Business account holder who is authorized to perform various API operations.” Ex.1006, 64. The merchant “must specify API credentials in each request to execute a PayPal API operation.” Ex.1006, 66; *see also* 67; Ex.1003, ¶¶150-153.

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

As discussed in [1.1.1]-[1.1.2], SetExpressCheckout and _ExpressCheckout teach the “*financial account registration function*”:



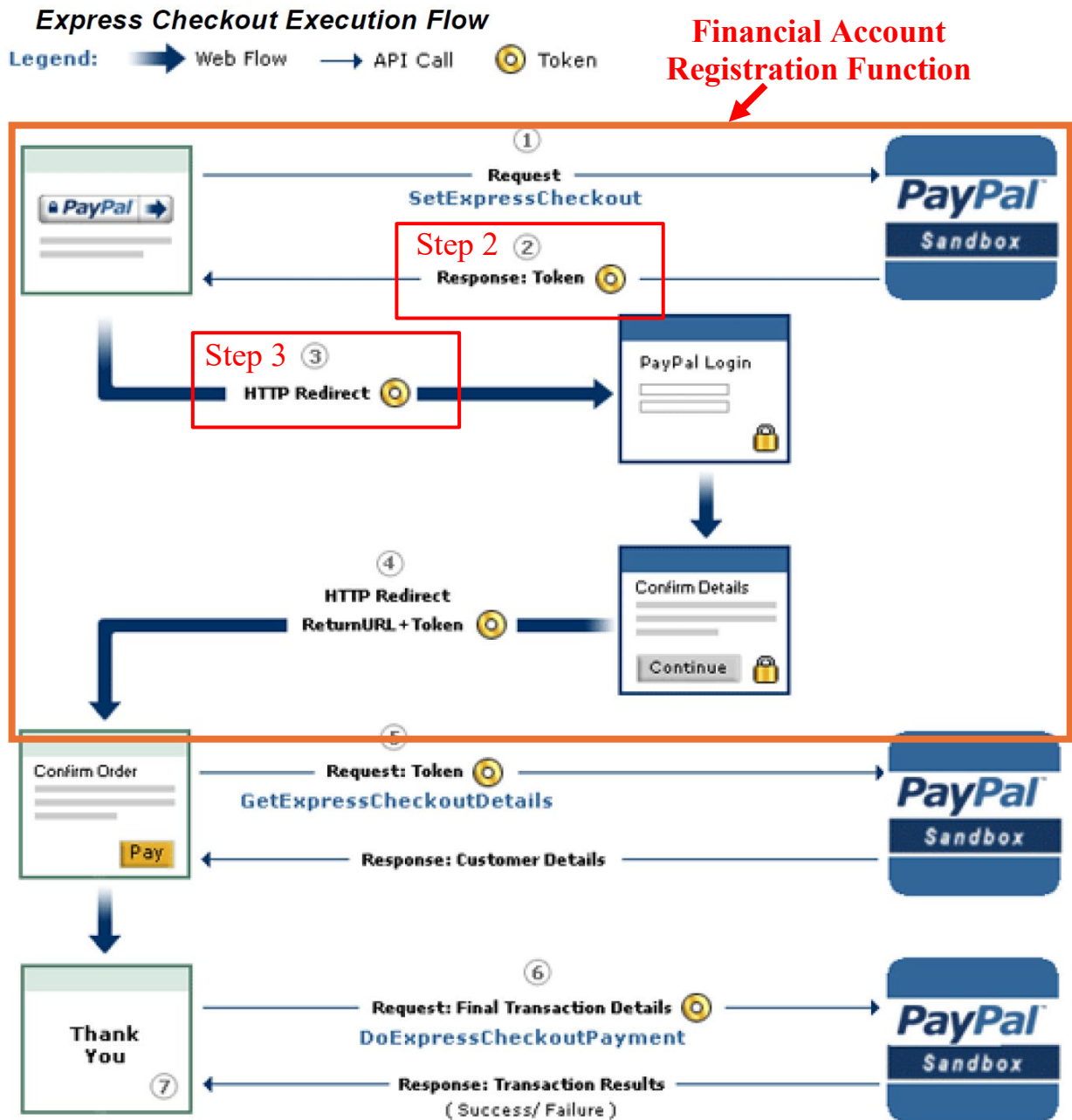
Ex.1006, 24 (annotated); Ex.1003, ¶154

The Express Checkout functions are executed upon being invoked by the merchant server. “To set up an Express Checkout transaction, [the merchant] must

invoke SetExpressCheckout... to provide sufficient information to initiate the payment flow and redirect to PayPal if the operation was successful.” Ex.1006, 19; *see also* 17 (“When [the merchant] redirect[s] [the] buyer’s browser to PayPal, [it] must specify the _ExpressCheckout command for Express Checkout.”). Ex.1003, ¶¶154-156.

[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,*

First, PayPal’s servers cause a URL to be generated by sending to the merchant a token that is to be combined with a static, predefined URL (“*generating a uniform resource locator*”). Ex.1003, ¶¶157-162. PayPal’s server sends to the merchant a “token that is used in subsequent steps” (Ex.1006, 25), as shown in Step 2 of the Example Transaction below:



Ex.1006, 24 (annotated); Ex.1003, ¶158

Sending the token at Step 2 then triggers Step 3. In Step 3, the merchant redirects the user's browser using a static URL, the token returned by `SetExpressCheckout`, and `_ExpressCheckout`. Ex.1006, 25 ("If the

[SetExpressCheckout] operation was successful, [the merchant] use[s] the token and redirect[s] [the] browser to the Sandbox to log in.”). Below is an example redirect:



The diagram shows a URL: `https://www.sandbox.paypal.com/cgi-bin/webscr?cmd=_express-checkout&token=EC-1NK66318YB717835M`. Red arrows point to specific parts of the URL with labels: 'HTTPS request' points to the 'https' protocol; 'URL' points to the entire URL; '_ExpressCheckout Command' points to the 'cmd=_express-checkout' parameter; and 'Token' points to the 'token=EC-1NK66318YB717835M' parameter. The entire URL is highlighted in yellow.

Ex.1006, 25 (annotated); Ex.1003, ¶161

Accordingly, PayPal’s servers generate a URL by sending the token in Step 2, which prompts the merchant server to combine that token with a static URL (“*generating a uniform resource locator (URL)*”). The static URL is described in further detail below with respect to [1.1.5.1b].

Second, the generated URL is used for “*establishing a secure socket layer connection via the internet between the secure server and the payer computing system.*” Ex.1003, ¶¶163-165. Communications to and from PayPal’s servers use a URL to establish an SSL connection between the sender and PayPal’s servers. *See* Ex.1006, 67-68 (“All requests to execute PayPal API operations sent via HTTP must be URL encoded.”), Ex.1007, 13 (“The PayPal SOAP API service is protected to ensure that only authorized PayPal members use it,” including “Secure Sockets Layer (SSL) data transport.”).

It was well-known in the art that HTTPS requests establish an SSL connection between the requesting browser and the destination server over the Internet. Ex.1003, ¶165; Ex.1014, 565 (“secure URLs will start with https:// instead of the standard (nonsecure) http://.”); Ex.1003, ¶¶157-166.

[1.1.5.1a] the URL comprising either: a dynamic URL generated by the secure server for the payer and the payee;

The claim language recites two alternative types of URLs—static or dynamic. Ex.1003, ¶167. As will be explained below in [1.1.5.1b], PayPal teaches a static URL as claimed and thus meets the claim. Nevertheless, it would have been obvious for PayPal to use a dynamic URL in place of a static URL, as was well-known in the art. Ex.1003, ¶¶167-168; *see* Ex.1019, 1 (explaining the difference between static and dynamic URLs); Ex.1016, [0058] (“The network address may be **static or dynamic**”); Ex.1018, [0046]; Ex.1017, [0036]. Thus, a POSITA would have understood how to modify PayPal so that PayPal’s server generates a dynamic URL rather using a static URL. Ex.1003, ¶¶167-169.

[1.1.5.1b] the URL comprising either: ... or a static URL and a hypertext transport protocol (HTTP) parameter used by the secure server to identify the payer and the payee;

PayPal’s example redirect includes a “*static URL*” and an “*HTTP parameter*.” As discussed in [1.1.5.1], the merchant redirects the buyer’s browser using a “*static URL*,” _ExpressCheckout, and the token (“*HTTP parameter*”)

returned by SetExpressCheckout. Below is an example redirect:

The image shows a URL: `https://www.sandbox.paypal.com/cgi-bin/webscr?cmd=_express-checkout&token=EC-1NK66318YB717835M`. Three red arrows point to parts of the URL with labels: one points to `https://www.sandbox.paypal.com/cgi-bin/webscr?` with the label *“static URL”*; another points to `cmd=_express-checkout` with the label *_ExpressCheckout Command*; and a third points to `&token=EC-1NK66318YB717835M` with the label *Token (“HTTP parameter”)*.

Ex.1006, 25 (annotated); Ex.1003, ¶170

As discussed in [1.1.3], the token “is used by other Express Checkout API operations and by the `_ExpressCheckout` command to identify the transaction,” and, thus, identifies the buyer and merchant (*“identif[ies] the payer and payee”*). Ex.1006, 18; *see also* 16 (“A token is a value assigned by PayPal that associates the execution of API operations and commands with a specific instance of a user experience flow.”). Because the token is used as part of the HTTP request along with the static URL, the token is used as an *“HTTP parameter”* as claimed.

Ex.1003, ¶171

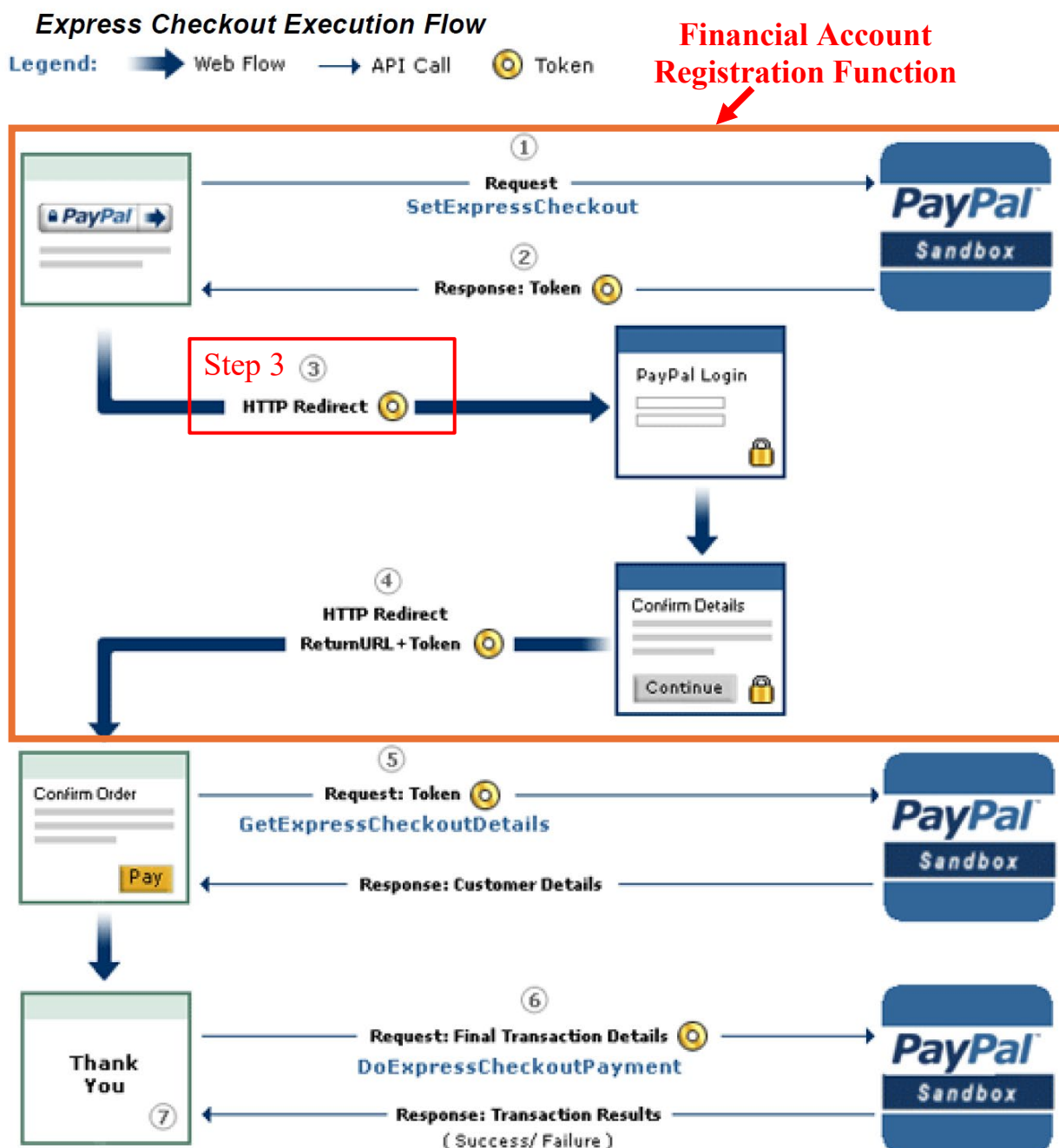
Moreover, as discussed in Section VI.C, a POSITA would have understood *“generating a... URL”* in [1.1.5.1] to include sending to the merchant a token to be combined with a predefined static URL. *See* Ex.1003, ¶¶71-86.

Thus, because PayPal discloses combining a static URL with the token identifying the transaction to redirect the buyer’s browser to PayPal’s website, PayPal renders this limitation obvious. Ex.1003, ¶¶170-173.

[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;*

First, Step 3 of the Example Transaction establishes an SSL connection between PayPal's servers and the buyer's browser ("*establishing the secure socket layer connection ... between the secure server and the payer computing system*").

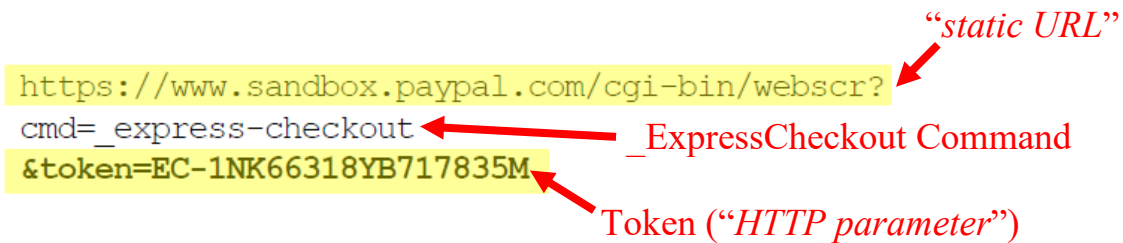
Ex.1003, ¶174; *see* [1.1.5.1]-[1.1.5.1b].



Ex.1006, 24 (annotated); Ex.1003, ¶174

Second, for the reasons explained above at [1.1.5.1], the SSL connection is established “in response to an HTTP request received from the merchant server for

the generated URL.” Ex.1003, ¶175. In other words, the HTTP request as defined by the static URL and token triggers the establishment of the SSL connection.

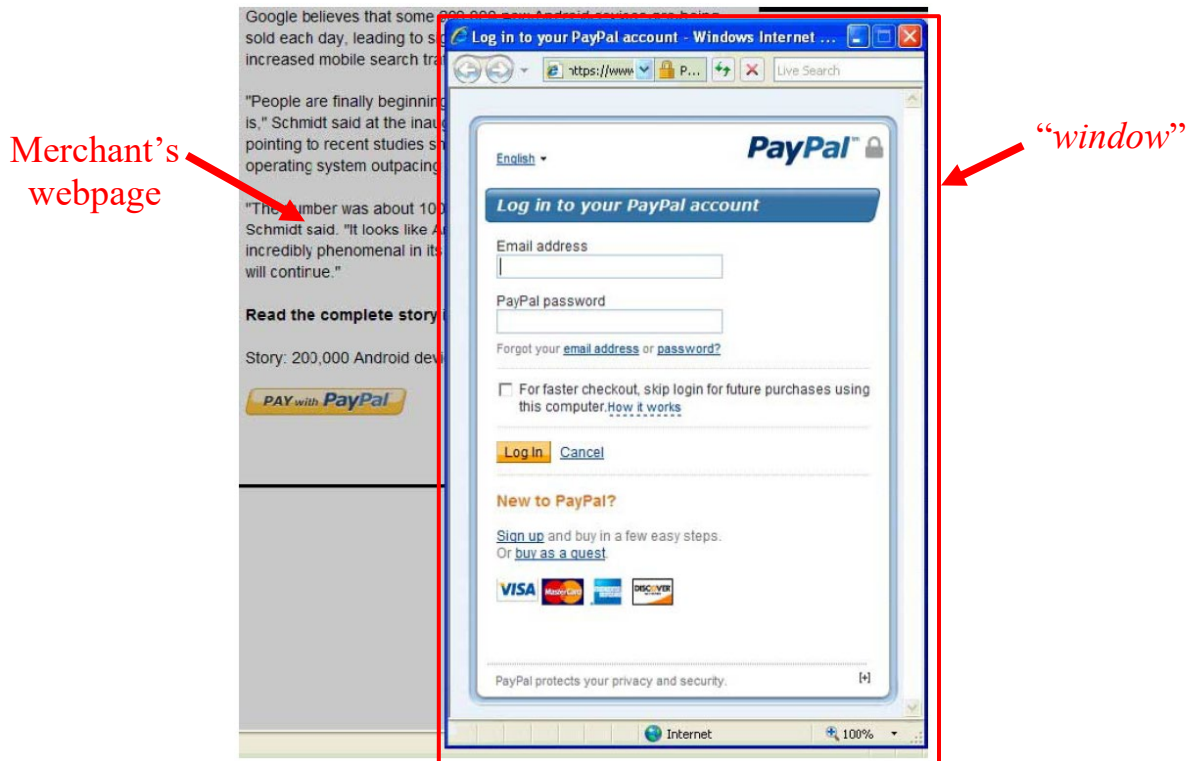


The diagram shows a URL with three parts highlighted in yellow and annotated with red arrows and text. The first part is `https://www.sandbox.paypal.com/cgi-bin/webscr?` with an arrow pointing to it from the label *“static URL”*. The second part is `cmd=_express-checkout` with an arrow pointing to it from the label *_ExpressCheckout Command*. The third part is `&token=EC-1NK66318YB717835M` with an arrow pointing to it from the label *Token (“HTTP parameter”)*.

```
https://www.sandbox.paypal.com/cgi-bin/webscr?  
cmd=_express-checkout  
&token=EC-1NK66318YB717835M
```

Ex.1006, 25 (annotated); Ex.1003, ¶175

Third, the secure connection is provided *“within a window or frame that is displayed within the webpage provided by the merchant server.”* Ex.1003, ¶¶176-177. When a user purchases digital goods, “[t]he digital goods payment process occurs in a lightbox that temporarily overlays [the merchant’s] website” such that “the digital goods payment process appears to be built into [the merchant’s] website.” Ex.1012, 10. When the buyer selects the PayPal payment option on the merchant’s website, the redirect “opens the Log in to your PayPal account page **in a mini-browser** [(“window”)], with a mask over the underlying page,” as shown in the screen shot below (Ex.1012, 11) (emphasis removed and added):



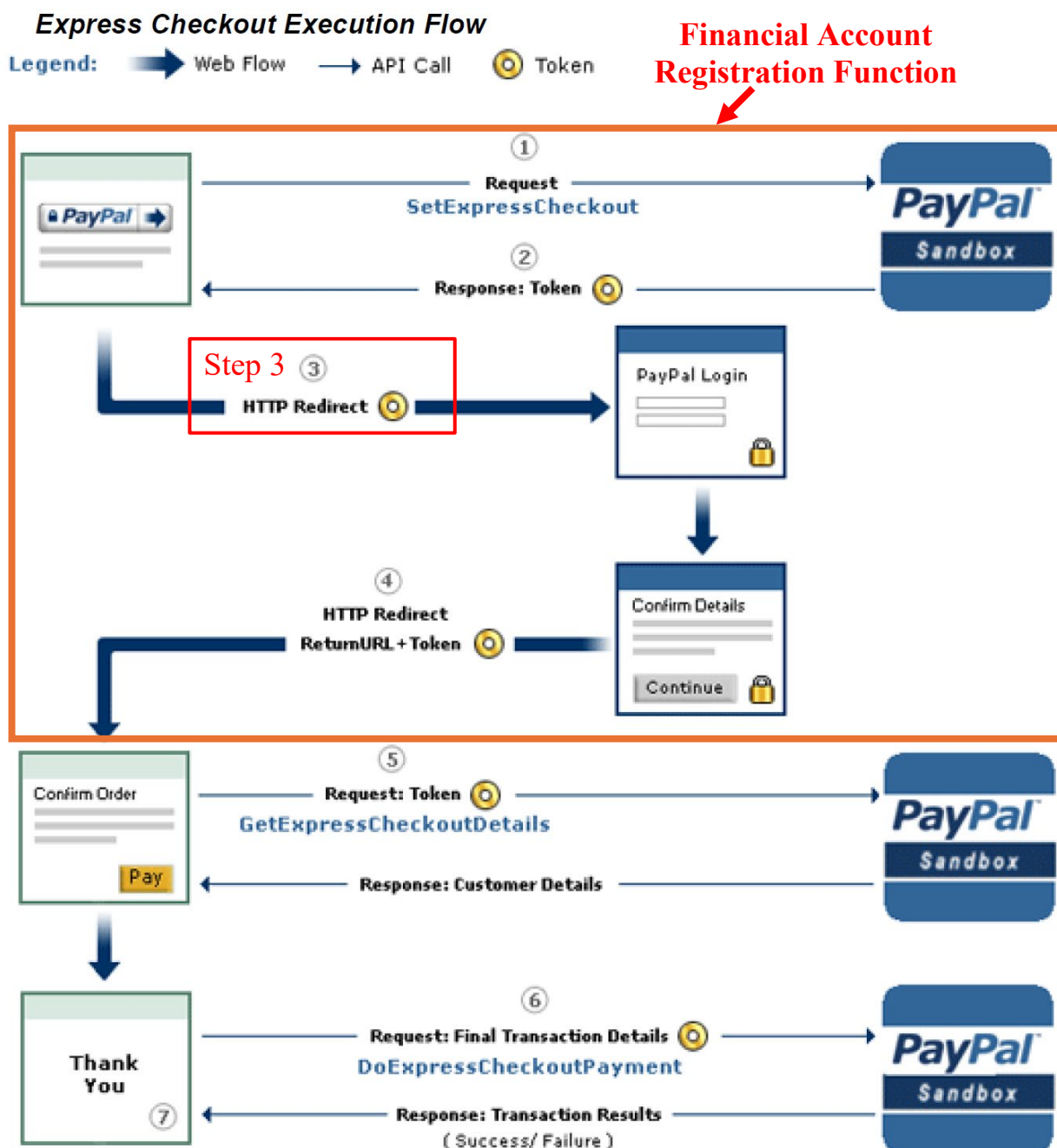
Ex.1012, 12; Ex.1003, ¶176

A POSITA would have found the alternatively claimed “*frame*” to be obvious as well. Ex.1003, ¶177. It was well-known in the art to use an inline frame to establish a connection between a buyer’s browser and a secure server. Ex.1003, ¶177; *see* Ex.1013, [0016] (“The payment system server 102 provides the business entity 104 with one or more **inline frames 122** that can be directly hosted on the website 120.”); Ex.1015, 8:54-57, 9:24-26. Therefore, a POSITA would have understood that PayPal could be modified such that the buyer’s browser can be redirected to PayPal’s website via an inline frame within the merchant’s website.

Ex.1003, ¶¶174-178.

[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*

First, as explained above at [1.1.5.2], PayPal’s server establishes an SSL connection to the buyer’s system in response to an HTTP request. Ex.1003, ¶179. Through that SSL connection, PayPal’s server provides instructions for displaying a window to the user (“*output[s] instructions to the payer computing system, in response to the HTTP request for the generated URL*”). Ex.1003, ¶¶179-182. As discussed in [1.1.5.1], the merchant redirects the buyer’s browser to PayPal’s website using a URL including an HTTPS request in Step 3 of the Example Transaction:

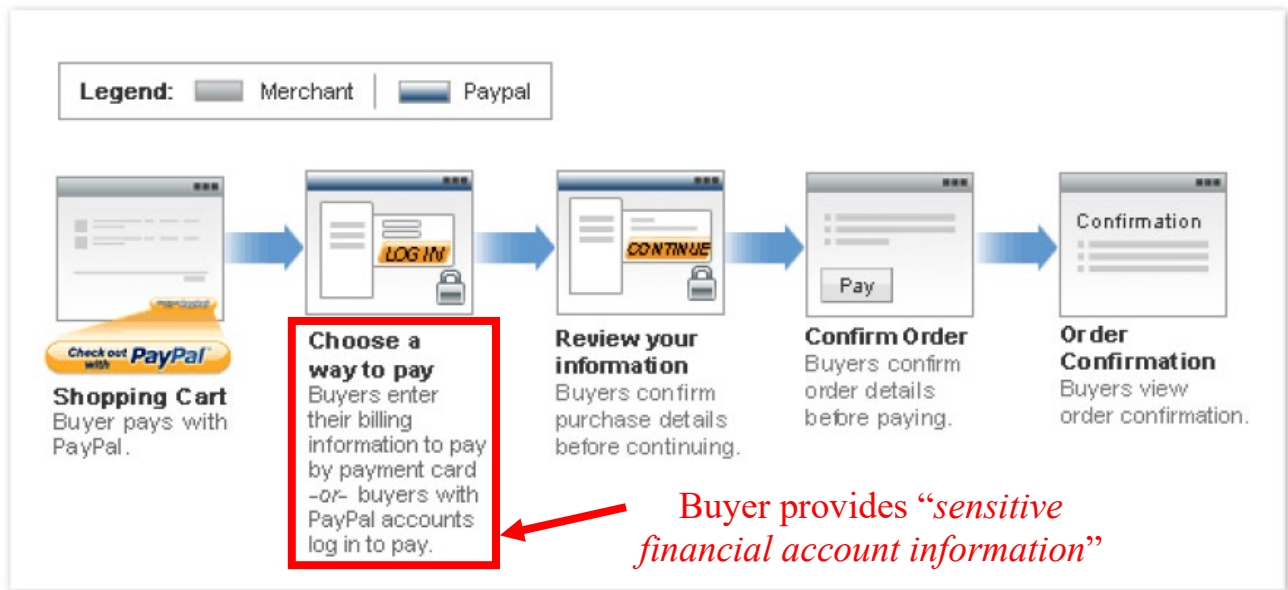


Ex.1006, 24 (annotated); Ex.1003, ¶179

The buyer's browser is directed to PayPal's log in page. *See* Ex.1006, 17 (“PayPal provides a command that [the merchant] use[s] when redirecting [its] buyer's browser to PayPal.”), 25 (“redirect [the] browser to the Sandbox to log

in.”). From the log in page, the buyer “[l]og[s] in to PayPal or choose[s] other ways to pay” (Ex.1009, 12):

The Express Checkout Experience



Ex.1009, 12 (annotated); Ex.1003, ¶181

Buyers can either “enter their billing information to pay by payment card” or log in to their PayPal account. Ex.1009, 12. Thus, in response to the HTTP request of the redirect URL (“*in response to the HTTP request for the generated URL*”), PayPal’s servers “*output instructions*” to render the PayPal log in page on the buyer’s browser (displayed on “*the payer computing system*”). Ex.1003, ¶185.

Second, PayPal’s website “*render[s] a financial account registration request form*” that “*provides functionality for the payer to provide sensitive financial account information associated with a financial account.*” Ex.1003, ¶¶-

185. PayPal’s log in page provides a “data entry area [that] lets buyers specify information about their payments.” Ex.1009, 10; *see also* Ex.1010, 20. Below is an example data entry form:

Example Data Entry Form in the PayPal Checkout User Interface

The screenshot displays the PayPal checkout user interface. It features a form titled "Choose a way to pay" in orange text. Below the title, there is a section titled "Pay with my PayPal account" in blue text, which is expanded. This section contains input fields for "Email" and "Password", a "LOG IN" button, and a link for "Forgot email address or password?". Below this section, there is a collapsed section titled "Don't have a PayPal account?" in blue text. This section contains the text "Pay with your debit or credit card as a PayPal guest" and a link for "Cancel and return to <Merchant name>". Red arrows point from the labels "Form title", "Form section, expanded", and "Form section, collapsed" to their respective elements in the form.

Ex.1009, 10; Ex.1003, ¶183

These forms allow the buyers to either “enter their PayPal credentials to log in and pay” or “**enter their billing information** and other contact information to pay with debit or credit cards.” Ex.1009, 11; *see also* Ex.1010, 22-23. Below is an example data entry form for a buyer who chooses to pay with a debit or credit card:

The screenshot shows a PayPal checkout interface. On the left, a 'Your order summary' box lists two items: 'Item description 1' for \$79.00 and 'Item description 2' for \$89.00. The total is \$202.04 USD. On the right, the 'Choose a way to pay' section shows 'Pay with my PayPal account' as the primary option. Below it, a link for 'Don't have a PayPal account?' leads to a credit card payment form. This form is highlighted with a red box and includes fields for Country (United States), Credit or debit card number, Expiration Date (mm / yy), CSC, First name, Last name, Billing Address Line 1, Billing Address Line 2 (Optional), Zip code, City, State (a dropdown menu), Phone, and Email Address. A red arrow points from the text 'Buyer provides "sensitive financial account information"' to the credit card form. At the bottom of the page, two red lines with dots point to the 'Cart review area' and the 'Data entry area'.

Buyer provides "sensitive financial account information"

Cart review area

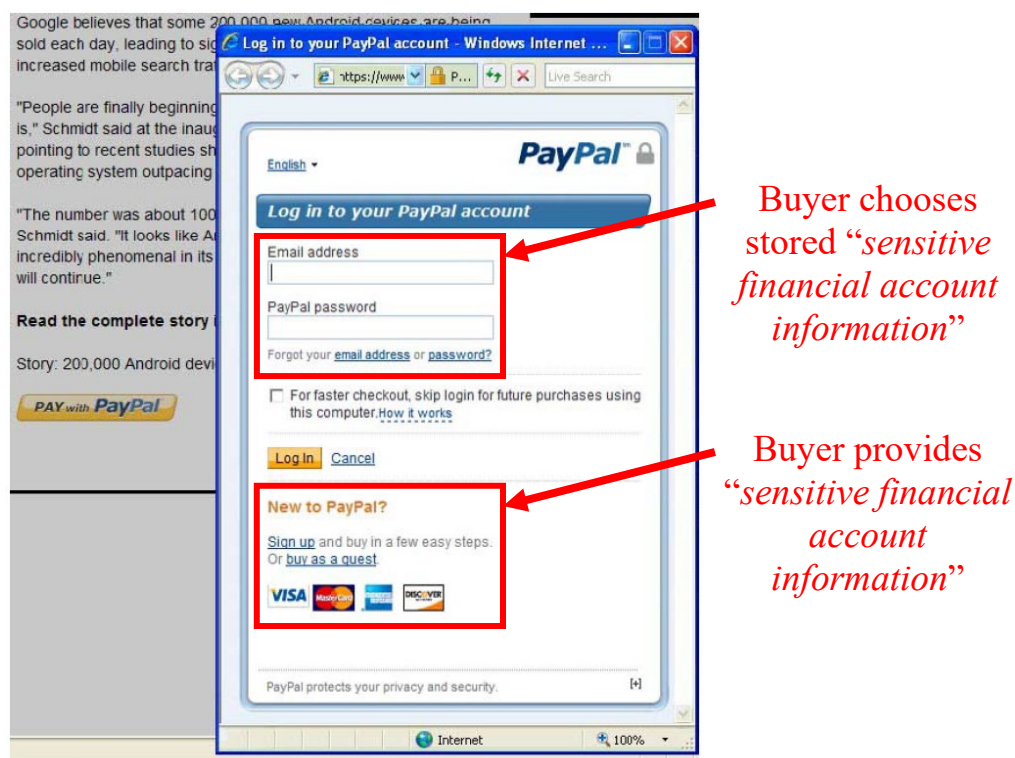
Data entry area

Ex.1009, 8 (annotated); Ex.1003, ¶184

Thus, the data entry forms (“*financial account registration request form*”) allows the buyer to input her financial/billing information (“*provides functionality for the payer to provide sensitive financial account information associated with a financial account*”). Ex.1003, ¶185.

Third, the registration form is “*within the window or frame that is displayed within the webpage provided by the merchant server.*” Ex.1003, ¶¶186-187. As

discussed in [1.1.5.2], PayPal discloses that the redirect occurs within a mini-browser overlaid on the merchant's website. For example, when purchasing digital goods, PayPal's log in page is rendered within the mini-browser:

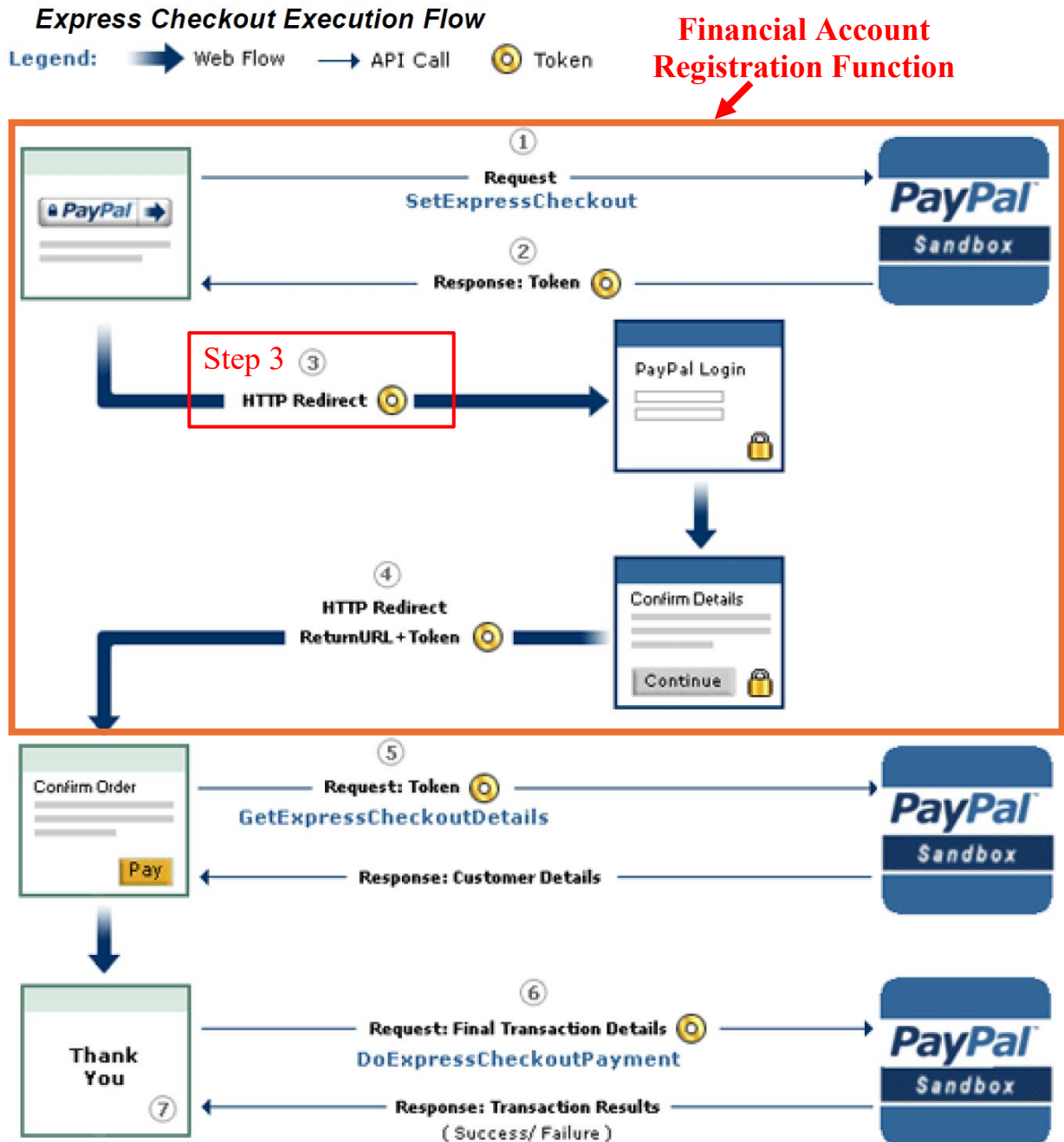


Ex.1012, 12 (annotated); Ex.1003, ¶186

From this page, "the buyer can log in" to her PayPal account or "[b]uyers without a PayPal account can sign up for one, or proceed as a guest and complete their purchase with a credit card." Ex.1012, 12. Thus, the data entry forms ("financial account registration request form") can be rendered within a mini-browser ("within the window or frame that is displayed within the webpage provided by the merchant server"). Ex.1003, ¶¶179-188.

[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;*

First, as discussed above at [1.1.5.3], PayPal’s servers output “*instructions to the payer computing system, in response to the HTTP request for the generated URL.*” Ex.1003, ¶¶189-190. As discussed in [1.1.5.3], the merchant redirects the buyer’s browser to PayPal’s website using a URL including an HTTPS request in Step 3:



Ex.1006, 24 (annotated); Ex.1003, ¶189

Moreover, as discussed in [1.1.5.3] a POSITA would have understood that the HTTPS request outputs instructions to the buyer's browser (displayed on the "payer computing system") that establishes an SSL connection between the buyer's

browser and PayPal's servers. Ex.1003, ¶190.

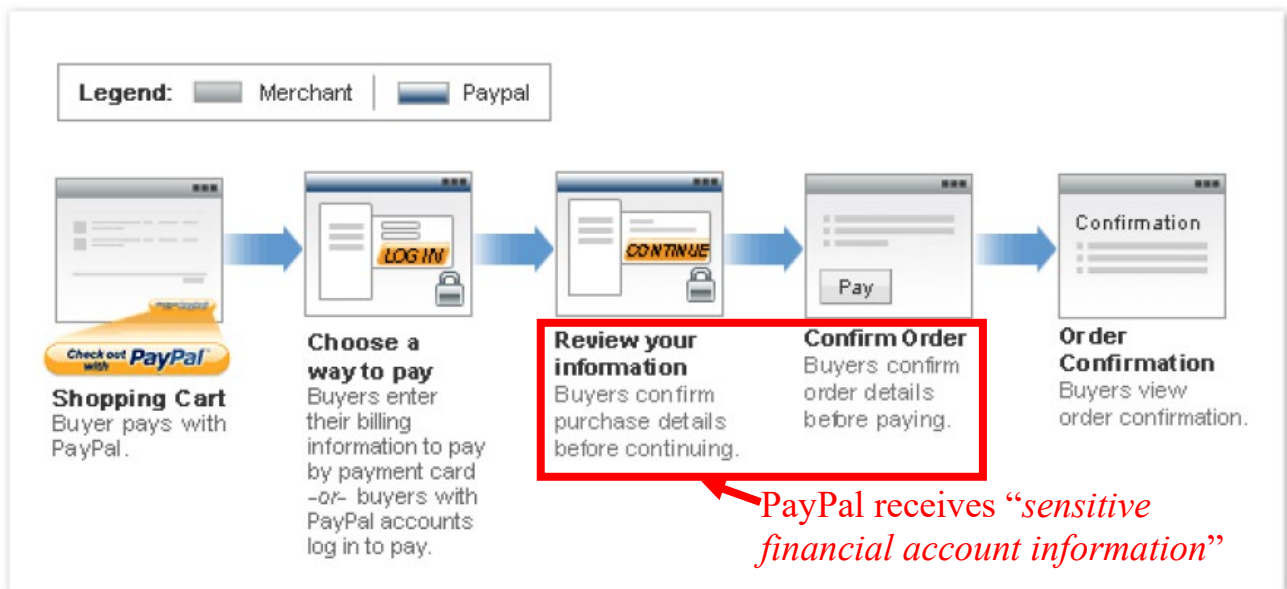
Second, the instructions “*encrypt the sensitive financial account information provided by the payer.*” Ex.1003, ¶191. When the SSL connection is established, “the Web server and Web browser **will encrypt** the HTTP messages before sending them across the Internet. Having the Web browser and server encrypt their transmissions protects your form results...from unauthorized access.” Ex.1014, 565. Additionally, a POSITA would have understood that communications sent via the SSL connection would be encrypted by the server. Ex.1003, ¶191.

Third, the instructions “*transmit the encrypted financial account information to the secure server via the secure socket layer connection.*” Ex.1003, ¶192. After the buyer inputs her billing/financial information (“*sensitive financial account information*”) into the data entry forms (“*financial account registration form*”), the buyer submits the information. Thus, the billing/financial information is sent to PayPal's servers (“*the secure server*”) via the SSL connection (see [1.2] below). See Ex.1006, 10 (“In the Express Checkout flow, the buyer” “Reviews the transaction on PayPal” and “Confirms the order and pays from [the merchant's] site.”); Ex.1009, 12; Ex.1010, 13. A POSITA would have understood that the message containing the billing/financial information (“*financial account information*”) would have been encrypted before sending over the SSL connection. Ex.1003, ¶¶189-193.

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

As discussed in [1.1.5.3], PayPal’s website provides data entry forms to the buyer’s browser that allow the buyer to provide her financial/billing information in Step 4 of the Example Transaction. After providing their financial/billing information by “[l]og[ging] in to PayPal or choos[ing] other ways to pay,” buyers “[r]eview their payments on PayPal” and “[c]onfirm their orders on [the merchant’s] website before paying” (Ex.1009, 12):

The Express Checkout Experience



Ex.1009, 12 (annotated); Ex.1003, ¶195

Thus, “[b]uyers enter and review information about their payments” via these forms provided by PayPal, then submit them to PayPal’s servers. See

Ex.1006, 10 (“In the Express Checkout flow, the buyer” “Reviews the transaction on PayPal” and “Confirms the order and pays from [the merchant’s] site.”);

Ex.1012, 13 (“If the buyer chooses Pay, the Thank you for using PayPal page appears in the mini-browser.”) (emphasis removed). Ex.1003, ¶¶194-197.

[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;*

First, PayPal’s servers store payment information received from a payer (“*storing the sensitive financial account information*”). Ex.1003, ¶¶198-199. The buyer’s “shipping, billing, or payment information” may be stored on their PayPal account. Ex.1006, 9; *see also* Ex.1009, 12. PayPal collects information including “[f]inancial information” such as “the full bank account numbers and/or credit card numbers that [the buyer] link[s] to [her] PayPal account or give[s] [PayPal] when [she] use[s] the PayPal Services.” Ex.1011, 1.

Second, the sensitive financial account information collected by PayPal is stored in “*a secure storage location*.” Ex.1003, ¶¶200-201. PayPal “use[s] computer safeguards such as firewalls and data encryption, enforce[s] physical access controls to [its] buildings and files, and authorize[s] access to personal information only for those employees who require it to fulfill their job responsibilities.” Ex.1011, 2. Additionally, as described in [1.1.4], PayPal only allows authenticated users to access PayPal’s servers and includes four levels of

security to prevent unauthorized access. Thus, PayPal’s servers, which store the billing/financial information associated with the buyer’s PayPal account, are a “*secure storage location*.” Ex.1003, ¶¶200-201.

Third, PayPal’s servers comply with federal and state regulations, thus “*performing each software process required to maintain compliance with one or more information security standards.*”⁷ Ex.1003, ¶202. PayPal’s servers, which store the billing/financial information, are protected “by maintaining physical, electronic and procedural safeguards in compliance with applicable US federal and state regulations.” Ex.1011, 2. The applicable US federal and state regulations are “*one or more information security standards.*” Ex.1003, ¶¶198-203.

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

As discussed in [1.1.1]-[1.1.2], GetExpressCheckoutDetails and DoExpressCheckoutPayment correspond to the “*token retrieval function*.” The merchant invokes GetExpressCheckoutDetails to retrieve the Payer ID (“*token*”) from PayPal’s servers and DoExpressCheckoutPayment using the Payer ID to

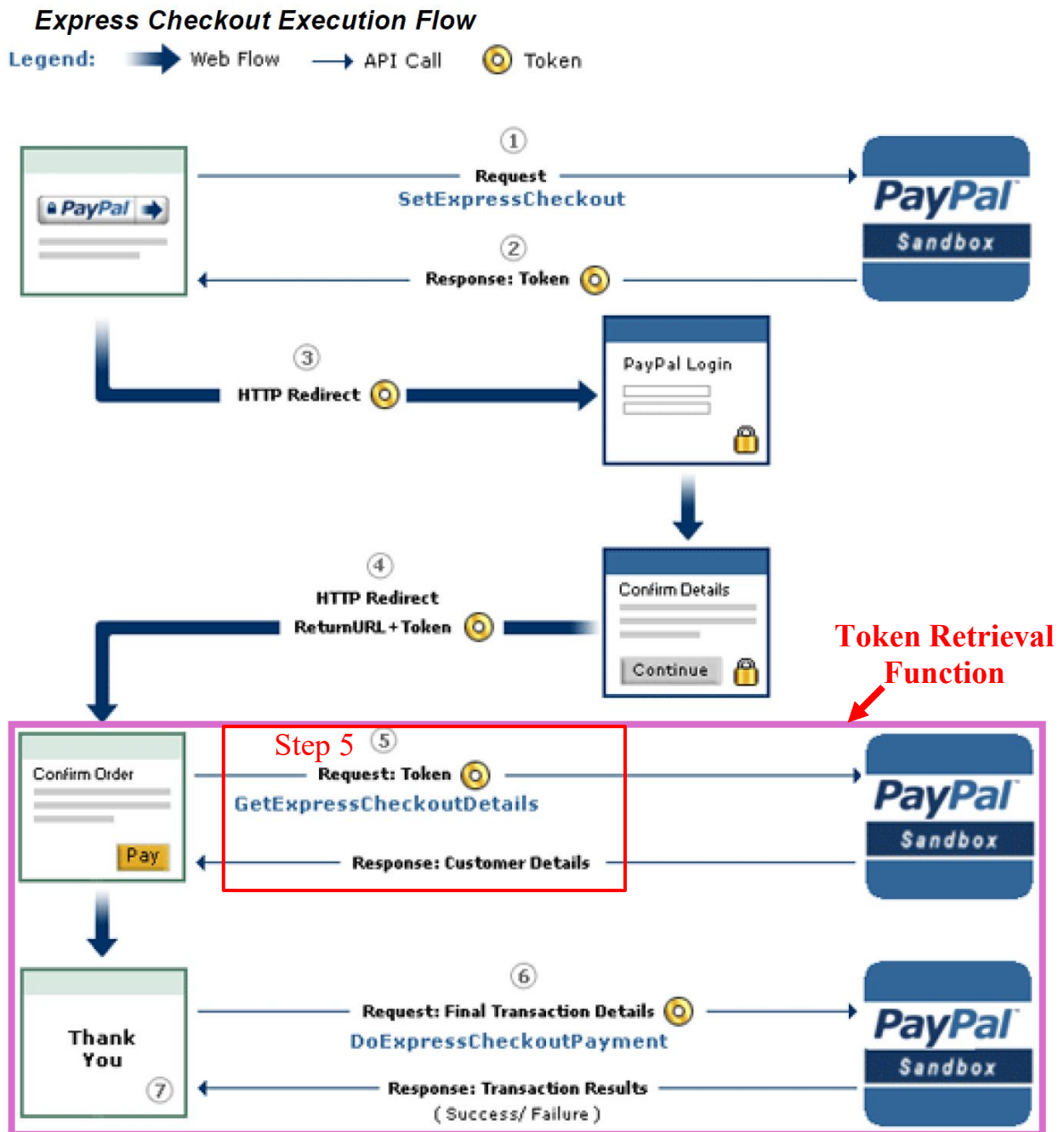
⁷ Petition does not concede that this term meets the definiteness requirements of 35 U.S.C. 112. While the scope of this term is unclear, it would at least include complying with federal and state regulations.

complete the transaction. *See* Ex.1006, 21 (“To obtain details about an Express Checkout transaction, [the merchant] can invoke the GetExpressCheckoutDetails API operation.”); 21 (“To complete an Express Checkout transaction, [the merchant] must invoke the DoExpressCheckoutPayment API operation.”).

Ex.1003, ¶¶204-206.

[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*

First, PayPal’s GetExpressCheckoutDetails “*provid[es] a non-sensitive electronic data token representing the sensitive financial account information to the merchant server.*” Ex.1003, ¶¶207-211. The merchant invokes GetExpressCheckoutDetails “[t]o obtain a buyer’s shipping address and Payer ID” (Ex.1006, 21) in Step 5 of the Example Transaction:



Ex.1006, 24 (annotated); Ex.1003, ¶208

Thus, the Payer ID (“*non-sensitive electronic data token*”) is returned to the merchant in the response to GetExpressCheckoutDetails, as shown in the below

example response:

```
TIMESTAMP=2007%2d04%2d05T23%3a44%3a11Z
&CORRELATIONID=6b174e9bac3b3
&ACK=Success
&VERSION=XX%2e000000
&BUILD=1%2e0006
&TOKEN=EC%2d1NK66318YB717835M
&EMAIL=YourSandboxBuyerAccountEmail
&PAYERID=7AKUSARZ7SAT8
&PAYERSTATUS=verified
&FIRSTNAME=...
&LASTNAME=...
&COUNTRYCODE=US
&BUSINESS=...
&PAYMENTREQUEST_0_SHIPTONAME=...
&PAYMENTREQUEST_0_SHIPTOSTREET=...
&PAYMENTREQUEST_0_SHIPTOCITY=...
&PAYMENTREQUEST_0_SHIPTOSTATE=CA
&PAYMENTREQUEST_0_SHIPTOCOUNTRYCODE=US
&PAYMENTREQUEST_0_SHIPTOCOUNTRYNAME=United%20States
&PAYMENTREQUEST_0_SHIPTOZIP=94666
&PAYMENTREQUEST_0_ADDRESSID=...
&PAYMENTREQUEST_0_ADDRESSSTATUS=Confirmed
```



Payer ID (“token”)

Ex.1006, 26 (annotated); Ex.1003, ¶210

The Payer ID is a “[u]nique PayPal customer account identification number.” Ex.1007, 100. Because the Payer ID identifies the customer’s account, including her financial/billing information, the Payer ID “*represent[s] the sensitive financial account information.*” Ex.1003, ¶211.

Second, the Payer ID is provided to the merchant server “*without providing the sensitive financial account information to the merchant server.*” Ex.1003, ¶212.

In the response to GetExpressCheckoutDetails, only the Payer ID (“token”), the

buyer's email, and shipping information is returned to the merchant. Ex.1003, ¶212. The Payer ID is a "[u]nique PayPal customer account identification number," such as, for example, 7AKUSARZ7SAT8. Ex.1007, 100. The Payer ID is an alphanumeric string that does not include financial/billing information. Thus, GetExpressCheckoutDetails does not provide financial/billing information ("*sensitive financial account information*") to the merchant. Ex.1003, ¶212; see Ex.1011, 3 ("Regardless, [PayPal] will not disclose [the buyer's] credit card number or bank account number to anyone [the buyer] ha[s] paid or who has paid [the buyer] through PayPal").

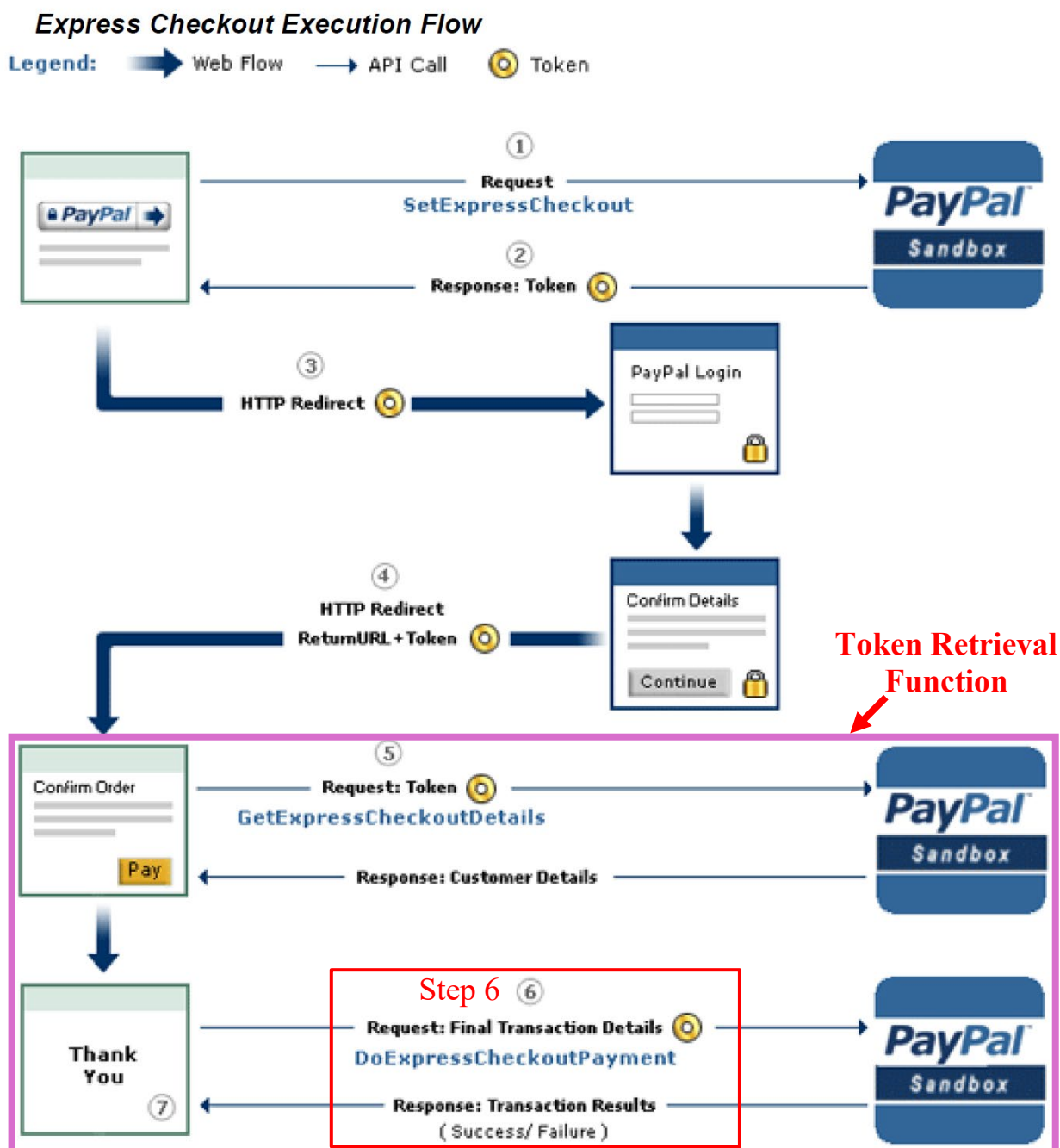
Third, the Payer ID is provided to the merchant server "*without providing the non-sensitive electronic data token to the payer.*" Ex.1003, ¶¶213-214. As discussed in Section VI.A., a POSITA would have understood that [1.4.1] merely requires the secure server provide the "*non-sensitive electronic data token to the merchant server...without providing the non-sensitive electronic data token to the payer*" **during execution of the token retrieval function.** See Ex.1003, ¶¶52-61. The response to GetExpressCheckoutDetails returns information about the payer, including the Payer ID to the merchant, **not** the buyer. See Ex.1006, 26. Thus, the Payer ID ("*token*") is not provided to the "*payer*" during execution of the token retrieval function. Ex.1003, ¶213.

A POSITA would have understood that this limitation does not apply to any

other step in the transaction process. Ex.1003, ¶214. Thus, regardless of whether PayPal provides the PayerID (“*non-sensitive electronic data token*”) to the payer at some other point, PayPal meets this limitation because it does not provide the PayerID to the payer during execution of the token retrieval function. Ex.1003, ¶214. Again, as mentioned above at VI.A, even the specification of the ’621 patent (parts incorporated by reference) allows for the non-sensitive electronic data token to be provided to the payer outside of the token retrieval function. Ex.1003, ¶60; *see* Ex.1005, 21, 31-32; Ex.1003, ¶¶215-207.

[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.*

First, the merchant invokes DoExpressCheckoutPayment with the Payer ID and a payment amount to complete the transaction (“*process[es] the payment transaction using the sensitive financial account information*”). Ex.1003, ¶¶216-220; Ex.1006, 21-23. DoExpressCheckoutPayment is Step 6 of the Example Transaction:



Ex.1006, 24 (annotated); Ex.1003, ¶217

Below is an example request invoking DoExpressCheckoutPayment:

```
<form method=post action=https://api-3t.sandbox.paypal.com/nvp>
  <input type=hidden name=USER value=API_username>
  <input type=hidden name=PWD value=API_password>
  <input type=hidden name=SIGNATURE value=API_signature>
  <input type=hidden name=VERSION value=XX.0>
  <input type=hidden name=PAYMENTREQUEST_0_PAYMENTACTION
    value=Authorization>
  <input type=hidden name=PAYERID value=7AKUSARZ78AT8>
  <input type=hidden name=TOKEN value= EC%2d1NK66318YB717835M>
  <input type=hidden name=PAYMENTREQUEST_0_AMT value= 19.95>
  <input type=submit name=METHOD value=DoExpressCheckoutPayment>
</form>
```

Payer ID
("token")

"payment
amount"

Ex.1006, 27 (annotated); Ex.1003, ¶218

Thus, the merchant provides the Payer ID and the payment amount to indicate to PayPal's servers that they should transfer the payment amount to the merchant using the financial/billing information ("*sensitive financial account information*") associated with the Payer ID.

Below is an example response from PayPal's servers the merchant invokes DoExpressCheckoutPayment:

```
TIMESTAMP=2007%2d04%2d05T23%3a30%3a16Z
&CORRELATIONID=333fb808bb23
ACK=Success
&VERSION=XX%2e000000
&BUILD=1%2e0006
&TOKEN=EC%2d1NK66318YB717835M
&PAYMENTREQUEST_0_TRANSACTIONID=043144440L487742J
&PAYMENTREQUEST_0_TRANSACTIONTYPE=expresscheckout
&PAYMENTREQUEST_0_PAYMENTTYPE=instant
&PAYMENTREQUEST_0_ORDERTIME=2007%2d04%2d05T23%3a30%3a14Z
&PAYMENTREQUEST_0_AMT=19%2e95
&PAYMENTREQUEST_0_CURRENCYCODE=USD
&PAYMENTREQUEST_0_TAXAMT=0%2e00
&PAYMENTREQUEST_0_PAYMENTSTATUS=Pending
&PAYMENTREQUEST_0_PENDINGREASON=authorization
&PAYMENTREQUEST_0_REASONCODE=None
```

Indicating transaction was successful

“payment amount”

Ex.1006, 27 (annotated); Ex.1003, ¶219

“If the operation was successful, the response should include ACK set to Success.” Ex.1006, 27. Thus, the response from DoExpressCheckoutPayment indicates that the transaction was processed successfully. Ex.1003, ¶220.

Therefore, the merchant specifying the Payer ID and the payment amount when invoking DoExpressCheckoutPayment to complete the transaction renders obvious “*processing the payment transaction using the sensitive financial account information.*” Ex.1003, ¶¶216-220.

Second, processing the payment transaction is done 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.*” Ex.1003, ¶¶221-222. PayPal

describes completing a transaction. Ex.1023, 7 (Instant Payment Notification “notif[ies] [the merchant] about events such as...[i]nstant payments, including Express Checkout,...and direct credit card payments.”); Ex.1024, 97 (“The status of the payment: ...Completed: The payment has been completed, and the funds have been added successfully to [the merchant’s] account.”). For example, the merchant’s PayPal account balance is credited when PayPal completes a credit card transaction from the buyer. *See* Ex.1025, 45 (“Monthly account statements list the totals of the payments received.”), 62 (“When [the merchant] receive[s] [a]...credit card transaction, the Balance Impact column indicates a credit.”). Accordingly, because PayPal receives the funds from the payer’s bank account, a POSITA would have understood that PayPal sent a request to the payer’s bank to obtain those funds. Ex.1003, ¶221.

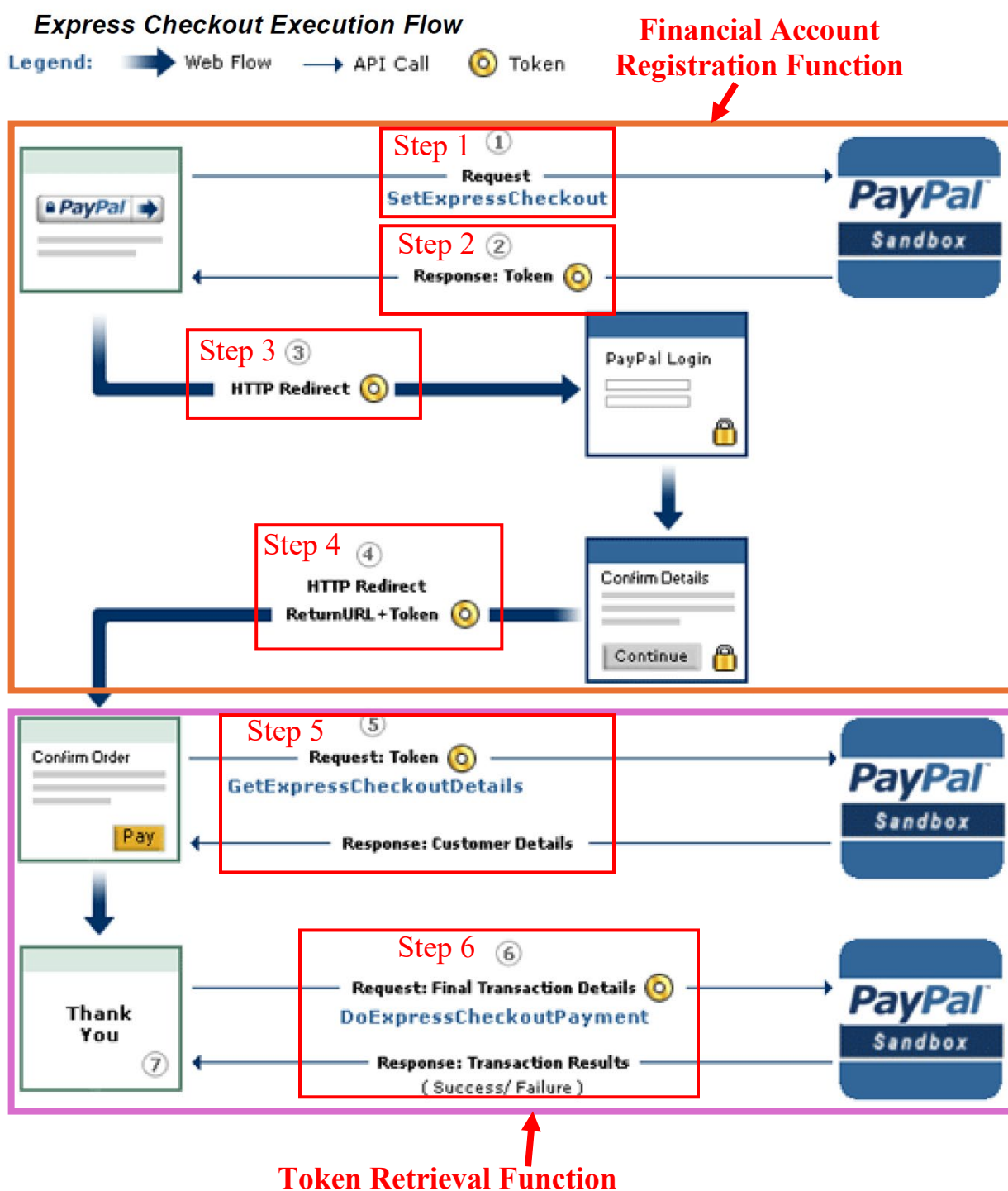
Indeed, it was well-known in the art to send an electronic request to the financial institution associated with the buyer’s financial/billing information to obtain the payment amount and forward it to the merchant. Ex.1003, ¶222; *see* Ex.1013, [0024] (“[A] portion of the monies held for the end user is transferred to the business entity utilizing the retrieved sensitive payment data via a backend payment gateway, 218”), [0014] (“The payment system server 102 is connected to a backend payment gateway 114 for processing transactions.”); Ex.1015, 23:5-13 (“All charges flagged for settlement are submitted in batches to the Processor (400)

and/or Card Network (500) directly. The Card Network (500) then causes the funds to be paid by the Card Issuing Bank (600) to Stripe (300) or to Stripe's Acquiring Bank (700). Stripe (300) then causes the settled funds to be sent to the Merchant (100) (or to the Merchant's Bank (800)), net of any applicable fees. The Card Issuing Bank (600) collects the paid funds from the Customer (200).”). Ex.1003, ¶¶216-223.

4. 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;*

First, PayPal teaches “*a second payment transaction instruction from the merchant server.*” Ex.1003, ¶¶224-225. As explained in claim 1 above, PayPal allows a merchant to collect a first payment from a buyer by invoking SetExpressCheckout to set up the Express Checkout transaction (Steps 1-2), redirecting the buyer’s browser to PayPal’s website so that the buyer can submit her financial/billing information to PayPal’s servers (Steps 3-4), invoking GetExpressCheckoutDetails to retrieve the Payer ID (Step 5), then invoking DoExpressCheckoutPayment to complete the transaction (Step 6), as shown in the Example Transaction below:



Ex.1006, 24 (annotated); Ex.1003, ¶224

A POSITA would have understood that a buyer could make a second

purchase on the merchant's website, which initiates the Example Transaction a second time. Ex.1003, ¶225. During this second process, the merchant sends a DoExpressCheckoutPayment request (“*a second payment transaction instruction from the merchant server*”) to PayPal's servers to request PayPal's servers transfer the second payment from the buyer to the merchant, as discussed in [1.4.2]. Ex.1003, ¶225.

Second, the DoExpressCheckoutPayment request “*includ[es] a representation of the non-sensitive electronic data token and specif[ies] a second payment amount.*” Ex.1003, ¶226. When invoking DoExpressCheckoutPayment, the merchant specifies the Payer ID (“*non-sensitive electronic data token*”) and the payment amount (“*second payment amount*”) to complete the transaction. Ex.1006, 21-23, 26-27. The Payer ID is a “*non-sensitive electronic data token*”) and, thus, is itself a “*representation of the non-sensitive electronic data token.*” Ex.1003, ¶¶224-227.

[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.*

As discussed in [1.4.2], in response to receiving the DoExpressCheckoutPayment request, PayPal's servers “*process[] the payment transaction using the sensitive financial account information*” by “*obtaining the*

payment amount, and forwarding at least a portion of the payment amount to the payee.” Ex.1003, ¶228.

As discussed in [2.1], a POSITA would have understood that the Payer ID would have been used to complete the second payment. *See* Ex.1003, ¶224-227. The Payer ID is a “[u]nique PayPal customer account identification number.” Ex.1007, 100; *see also* Ex.1006, 17 (“The buyer’s unique identifier (Payer ID)”). A POSITA would have understood that, if the same buyer makes a second purchase on the merchant’s website using PayPal, the same unique PayPal customer account identification number (Payer ID) will be used to identify the buyer that was used for the first purchase. Ex.1003, ¶229. Thus, it would have been obvious to a POSITA that PayPal’s servers would use the financial account information represented by the Payer ID to complete the second payment. Ex.1003, ¶¶228-230.

5. 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.*

As discussed in [2.1], a POSITA would have understood that the second payment is collected using the Payer ID (“*non-sensitive electronic data token*”) itself, rendering this limitation obvious. Ex.1003, ¶231.

6. 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.

As discussed in [1.4.1], the Payer ID represents the buyer's financial/billing information. The Payer ID is a "[u]nique PayPal customer account identification number" such as 7AKUSARZ7SAT8. Ex.1007, 100; Ex.1006, 26. Thus, because the Payer ID is an alphanumeric customer account identification number that represents the customer's account—including the financial/billing information—the Payer ID ("*non-sensitive electronic data token*") is a "*symbolic representation*" of the financial/billing information ("*financial account information*"). Ex.1003, ¶232.

7. 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.*

As discussed in [1.1.3], the merchant specifies the token ("*at least one data element associated with the payer*") and a payment amount when invoking the DoExpressCheckoutPayment. The token "associates the execution of API operations and commands with a specific instance of a user experience flow." Ex.1006, 16. The merchant may set the value of the token when it calls SetExpressCheckout to set up the transaction: "If [**the merchant**] **set[s] the token in the SetExpressCheckout request**, the value of the token in the response is identical to the value in the request." Ex.1007, 91; Ex.1003, ¶233.

A POSITA would have understood that the merchant can set any value as the token. Ex.1003, ¶234. For example, a POSITA would have found it obvious that the merchant would set the token value to be a customer identification number generated by the merchant so that the transaction can be tied to a particular buyer. Ex.1003, ¶¶233-236, Ex.1013, [0015] (“[A] unique customer identifier created by the merchant could also be stored for the customer.”).

8. 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.*

First, PayPal discloses “a process, performed by the merchant server, to receive enrollment data provided by the payer.” Ex.1003, ¶¶237-241.

GetExpressCheckoutDetails allows a merchant to “obtain[] information about the buyer from PayPal, including shipping information” and any “populated fields are returned in the response.” Ex.1006, 17; 21, *see also* 21 (The merchant can invoke GetExpressCheckoutDetails “to obtain information about the buyer.”).

PayPal provides dozens of fields that can be included in the response to GetExpressCheckoutDetails. *See* Ex.1007, 92-108 (“GetExpressCheckoutDetails Response Message”). For example, the response may include “[i]nformation about the payer” (“PayerInfo”), the “[p]ayer’s contact telephone number” (“ContactPhone”), “[t]he buyer’s email address” (“BuyerMarketingEmail”), the

payer’s name (“PayerNameTypeFields”), or the shipping address (“AddressTypeFields”):

PayerInfo	ebl:PayerInfoType Information about the payer.
-----------	---

Ex.1007, 98; Ex.1003, ¶238

ContactPhone	xs:string Payer’s contact telephone number. NOTE: PayPal returns a contact telephone number only if your Merchant account profile settings require that the buyer enter one. Character length and limitations: Field mask is XXX-XXX-XXXX (for US numbers) or +XXX XXXXXXXXX (for international numbers)
--------------	--

Ex.1007, 99; Ex.1003, ¶238

BuyerMarketingEmail	ebl:EmailAddressType The buyer email address opted in by the buyer on the PayPal Review page. Limitations: 127 single-byte characters
---------------------	---

Ex.1007, 100; Ex.1003, ¶238

PayerNameType Fields

Field	Description
Salutation	<code>xs:string</code> Payer's salutation. Character length and limitations: 20 single-byte characters.
FirstName	<code>ebl:PersonNameType</code> Payer's first name. Character length and limitations: 25 single-byte characters.
MiddleName	<code>ebl:NameUser</code> Payer's middle name. Character length and limitations: 25 single-byte characters.
LastName	<code>ebl:NameType</code> Payer's last name. Character length and limitations: 25 single-byte characters.
Suffix	<code>ebl:SuffixType</code> Payer's suffix. Character length and limitations: 12 single-byte characters.

Ex.1007, 101; Ex.1003, ¶238

AddressType Fields

Field	Description
Name	<code>xs:string</code> Person's name associated with this shipping address. Character length and limitations: 32 single-byte characters.

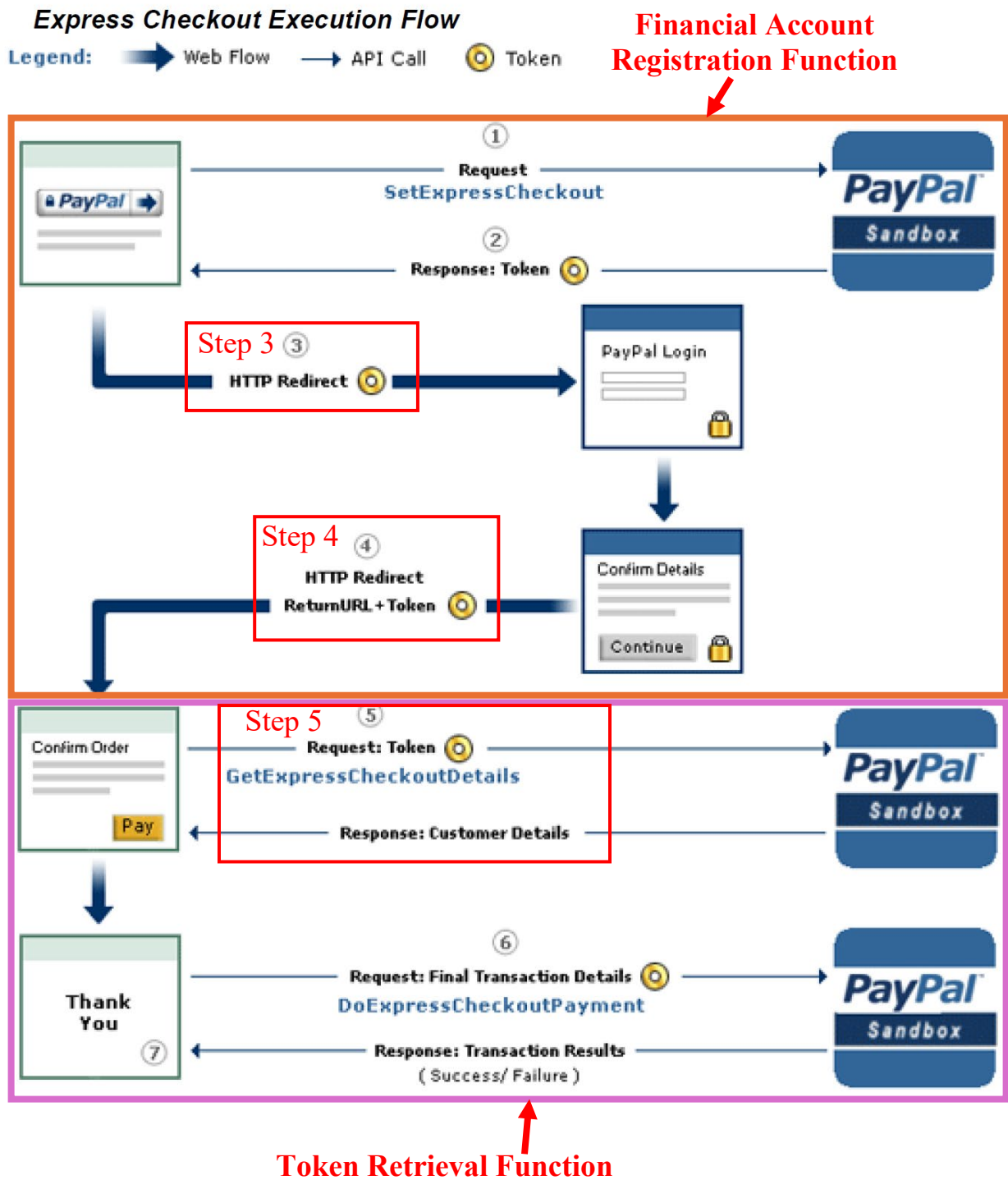
Field	Description
Street1	xs:string First street address. Character length and limitations: 100 single-byte characters.
Street2	xs:string Second street address. Character length and limitations: 100 single-byte characters.
CityName	xs:string Name of city. Character length and limitations: 40 single-byte characters.
StateOrProvince	xs:string State or province. Character length and limitations: 40 single-byte characters.
PostalCode	xs:string U.S. zip code or other country-specific postal code. Character length and limitations: 20 single-byte characters.
Country	ebl:CountryCodeType Country code. Character limit: 2 single-byte characters.
Phone	xs:string Phone number. Character length and limit: 20 single-byte characters.
AddressStatus	ebl:AddressStatusTypeCode Status of street address on file with PayPal. Valid values are: • none • Confirmed • Unconfirmed

Ex.1007, 101-102; Ex.1003, ¶238

As indicated by the '621 patent, enrollment data can include any data that can be used to enroll a buyer in an account with the merchant or the secure server (e.g. PayPal). *See* Ex.1001, 2:64-3:3 (“The token is an index to the stored financial account information for that customer that is stored by the merchant...**in connection with the customer record and is used to identify the customer**”).

account to be debited to process a payment authorized by the customer.”); *see also* 3:9-15. Thus, any of the information described above returned by GetExpressCheckoutDetails that identifies the buyer (*e.g.*, buyer’s phone number, name, email, and/or shipping address) is enrollment data that a merchant can use to create a customer account for the buyer. Ex.1003, ¶¶237-241; *see* Ex.1006, 69 (“[The merchant] can log other information that may be useful” such as “the entire request and response.”).

Second, the “*secure socket layer*” is “*asynchronous*” with GetExpressCheckoutDetails. Ex.1003, ¶¶242-244. The merchant “invoke[s] the GetExpressCheckoutDetails API operation from the page specified by return URL, which [the merchant] set in [its] call to the SetExpressCheckout API.” Ex.1006, 21. The merchant typically invoke[s] GetExpressCheckoutDetails “as soon as the redirect occurs and use[s] the information in the response to populate [its] review page” (Ex.1006, 21) in Step 5 of the Example Transaction:



Ex.1006, 24 (annotated); Ex.1003, ¶242

As discussed in [1.1.5.1], when the merchant redirects the buyer's browser

to PayPal's website in Step 3, an SSL connection is established between the buyer's browser and PayPal's servers. After the buyer confirms her financial/billing information, PayPal's servers redirect the buyer's browser back to the merchant's website in Step 4 of the Example Transaction using the return URL the merchant specified in SetExpressCheckout. Ex.1006, 25 ("When [the merchant testing Express Checkout] confirm[s], the Sandbox redirects [the] browser to the return URL [the merchant] specified when invoking the SetExpressCheckout API operation."); *see also* Ex.1006, 20.

Thus, if the merchant invokes GetExpressCheckoutDetails as soon as the redirect occurs in Step 4, the process of receiving enrollment data is asynchronous with the SSL connection. Ex.1003, ¶¶237-245.

9. 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.

As discussed in limitations [1.1.5.3] and [1.2], PayPal's website provides data entry forms for a buyer to provide her financial/billing information and submit it to PayPal's servers. These data entry forms allow buyers to "enter their billing information...to pay with debit or **credit cards**." Ex.1009, 11; Ex.1003, ¶¶246-249 *see also* Ex.1010, 22-23; Ex.1012, 12. Below is an example data entry form:

The screenshot shows a PayPal checkout interface. On the left, a 'Your order summary' box is outlined in blue and labeled 'Cart review area' with a red line. It lists two items: 'Item description 1' for \$79.00 and 'Item description 2' for \$89.00, with a total of \$202.04 USD. On the right, the 'Choose a way to pay' section is labeled 'Data entry area' with a red line. It features a 'Pay with my PayPal account' button and a 'Don't have a PayPal account?' section. The latter is highlighted with a red box and labeled 'credit card number' with a red arrow. This section includes fields for Country (United States), Credit or debit card number, Expiration Date (mm / yy), and CSC. Below these are fields for First name, Last name, Billing Address Line 1, Billing Address Line 2 (Optional), Zip code, City, State (a dropdown menu), and Phone. An 'Email Address' field is also present. A 'CONTINUE' button is at the bottom right of the form. A red line points from the 'Data entry area' label to the 'CONTINUE' button. At the bottom left, there is a link to 'PayPal protects your privacy and security. For more information, read our [Use Agreement](#) and [Privacy Policy](#).'

Description	Amount
Item description 1	\$79.00
Item # 123	
Item Price: \$79.00	
Qty: 1	
Item description 2	\$89.00
Item # 123	
Item Price: \$89.00	
Qty: 1	
Item total	\$168.00
Tax	\$15.54
Shipping and handling	\$18.50
Total	\$202.04 USD

Country: United States

Credit or debit card number

Expiration Date: mm / yy

CSC

What's this?

First name

Last name

Billing Address Line 1

Billing Address Line 2 (Optional)

Zip code

City

State: Select

e.g. 555-555-5555

Phone

Email Address

This transaction will appear on your statement as: [Merchant name](#)

CONTINUE

Secure checkout by **PayPal**

Cancel and return to: [Merchant name](#)

PayPal protects your privacy and security. For more information, read our [Use Agreement](#) and [Privacy Policy](#).

Ex.1009, 8 (annotated); Ex.1003, ¶246

10. Claim 15

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

As discussed in [1.1], the merchant's server hosts its website and communicates with PayPal's server via PayPal's SOAP API.

As discussed in [6.1], the merchant invokes GetExpressCheckoutDetails to

retrieve the Payer ID and other information about the payer, including the buyer's phone number, name, email, and address and it would be obvious to a POSITA that the merchant server could create a customer account for the buyer by storing the information about the buyer. Ex.1003, ¶252.

The merchant “can log other information that may be useful” including “implement[ing] a scheme that logs the entire request and response.” Ex.1006, 69. For example, the merchant may receive an instant payment notification (“IPN”) “when [it] invoke[s] an API operation, such as DoExpressCheckoutPayment.” Ex.1023, 11. When the merchant receives the IPN, the merchant may “store information in a customer relationship management (CRM) or accounting system.” Ex.1023, 11. Thus, PayPal contemplates storing information, such as a response including enrollment data, in a CRM system for the buyer. Ex.1003, ¶253. Therefore, the merchant receiving information about the buyer that can be used for to create a customer account for the buyer renders obvious this limitation. Ex.1003, ¶¶250-254.

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

See [1.1]. Ex.1003, ¶255.

[15.2] *receiving enrollment data identifying the payer;*

See [6.1]. Ex.1003, ¶¶256-257.

[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 [1.0] and [1.1.3]. Ex.1003, ¶258.

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

See [1.1.5.2]. Ex.1003, ¶259.

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

As explained at [1.1.4], PayPal authenticates the payee using a username and password (“*authentication credentials*”) to access the API. The username and password both identify and authenticate the payee. Ex.1003, ¶¶260-261.

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

As discussed in [1.1.5], the merchant initiates the “*financial account registration function*” by invoking SetExpressCheckout and redirecting the buyer’s browser with _ExpressCheckout. The merchant “must specify API credentials in each request to execute a PayPal API operation.” Ex.1006, 66. Thus, when invoking SetExpressCheckout, the merchant provides its credentials to authenticate that it is making the request and thereby establishes an “*authenticated session*” for the transaction. Ex.1003, ¶¶262-263.

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

See [1.1.5.2]. Ex.1003, ¶264.

[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 [1.1.5.3]. Ex.1003, ¶265.

[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;*

As discussed in [1.1.5.4], the SSL connection between the buyer's browser and PayPal's servers encrypts all messages between them.

Moreover, after redirecting the buyer's browser to PayPal, "[i]f the buyer approves the payment, PayPal redirects to the success URL with the following information:" "[t]he token that was included in the redirect to PayPal" and "[t]he buyer's unique identifier (Payer ID)." Ex.1006, 17. Because neither the token nor the Payer ID includes financial/billing information, no financial/billing information is provided to the merchant during the financial account registration function ("*without providing the sensitive financial account information to the merchant server*"). Ex.1003, ¶267; see [1.4.1].

Therefore, the SSL connection encrypting messages between the buyer's browser and PayPal's servers without sending the financial/billing information to

the merchant, PayPal renders obvious this limitation. Ex.1003, ¶¶266-268.

[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;*

As discussed in [1.4], the merchant initiates the token retrieval function by invoking GetExpressCheckoutDetails. Moreover, as discussed in [1.4.1] and [1.4.2], the GetExpressCheckoutDetails response includes the Payer ID (“*non-sensitive electronic data token*”).

The merchant “must specify API credentials in each request to execute a PayPal API operation.” Ex.1006, 66. Thus, the merchant initiates the token retrieval function during the “*authenticated session*.” Ex.1003, ¶¶269-271.

[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 [1.4.1]. Ex.1003, ¶272.

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

As discussed in [6.1], the GetExpressCheckoutDetails response can include information about the buyer, such as the buyer’s phone number, email, name or address, which can be considered “*enrollment data identifying the payer*.”

PayPal explains that the merchant “should log basic information about each

PayPal API operation [it] execute[s].” Ex.1006, 69. For example, the merchant may “implement a scheme that logs the entire request and response.” Ex.1006, 69. Thus, the merchant may log the entire GetExpressCheckoutDetails response, which includes the Payer ID (“*token*”) and the information about the buyer (“*enrollment data identifying the payer*”) together. Moreover, the merchant may receive an IPN “when [it] invoke[s] an API operation.” Ex.1023, 11. Thus, when the merchant receives the IPN after invoking GetExpressCheckoutDetails, the merchant may “store information in a customer relationship management (CRM) or accounting system,” like the Payer ID and the information about the buyer. Ex.1023, 11.

Ex.1003, ¶¶273-276.

[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 [1.5.2]. Ex.1003, ¶277.

11. Claim 16

[16.0] The method of claim 15, wherein ... data token.

See [3.1]. Ex.1003, ¶278.

12. Claim 17

[17.1] The method of claim 15, wherein ... account information.

See [4.1]. Ex.1003, ¶279.

13. Claim 18

[18.1] *The method of claim 15, wherein ... the payer.*

See [5.1]. Ex.1003, ¶280.

14. Claim 19

[19.1] *The method of claim 15, wherein ... resource locator (URL).*

See [1.5.1.1a]-[1.1.5.1b]. Ex.1003, ¶281.

15. Claim 20

[20.1] *The method of claim 15, wherein ... the payer.*

See [6.1]. Ex.1003, ¶282.

16. Claim 21

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

As discussed in [1.1.5.1], the merchant redirects the buyer's browser to PayPal's website, thereby establishing an SSL connection between the buyer's browser and PayPal's servers. *See* Ex.1006, 17 ("PayPal provides a command that [the merchant] use[s] when redirecting [its] buyer's browser to PayPal. This command enables [the] buyer to log into PayPal to approve an Express Checkout payment.").

As explained in [6.1], GetExpressCheckoutDetails returns information about the buyer such as her phone number, email, name or address.

As explained in limitations [1.1.5.3] and [1.2], the buyer provides and

submits her financial/billing information via a data entry form. Below is an example data entry form:

The image shows a screenshot of a PayPal checkout page. On the left is a 'Your order summary' table. On the right is a 'Choose a way to pay' section. A red box highlights the 'Don't have a PayPal account?' section, which contains fields for Country, Credit or debit card number, Expiration Date, CSC, Why's this?, First name, Last name, Billing Address Line 1, Billing Address Line 2 (Optional), Zip code, City, State, Phone, and EmailAddress. Red arrows point from text labels to these fields: 'sensitive financial account information' points to the Country dropdown; 'Name' points to the First and Last name fields; 'Address' points to the Billing Address Line 1 and 2 fields; 'Phone' points to the Phone field; and 'Email' points to the EmailAddress field. A 'CONTINUE' button is at the bottom of the form. A 'Cancel and return to <Merchant name>' link is at the bottom left. A 'PayPal protects your privacy and security' notice is at the bottom left. A 'Secure checkout by PayPal' logo is at the bottom right.

Description	Amount
Item description 1	\$79.00
Item # 123	Item Price: \$79.00
Qty: 1	
Item description 2	\$89.00
Item # 123	Item Price: \$89.00
Qty: 1	
Item total	\$168.00
Tax	\$15.54
Shipping and handling	\$18.50
Total	\$202.04 USD

Choose a way to pay

Pay with my PayPal account

Don't have a PayPal account?

Pay with your debit or credit card as a PayPal guest

Country: United States

Credit or debit card number

Expiration Date: mm / yy

CSC

Why's this?

First name

Last name

Billing Address Line 1

Billing Address Line 2 (Optional)

Zip code

City

State: Select

Phone

EmailAddress

CONTINUE

Cancel and return to <Merchant name>

PayPal protects your privacy and security. For more information, read our [User Agreement](#) and [Privacy Policy](#).

Secure checkout by PayPal

Cart review area

Data entry area

"sensitive financial account information"

Name

Address

Phone

Email

Ex.1009, 8 (annotated); Ex.1003, ¶284

This example form allows the buyer to provide her phone number, email, name or address ("enrollment data") over the internet ("network"). Ex.1003, ¶¶283-286.

17. Claim 22

[22.1] *The method of claim 15, wherein ... account number.*

See [7.1]. Ex.1003, ¶287.

18. Claim 25

[25.0] *A method ... the method comprising:*

See [1.0]. Ex.1003, ¶288.

[25.1] *providing, ... (API) that:*

See [1.1]. Ex.1003, ¶289.

[25.1.1] *provides ... transaction;*

See [1.1.1]. Ex.1003, ¶290.

[25.1.2] *provides ... server;*

See [1.1.2]. Ex.1003, ¶291.

[25.1.3] *receives, ... payee;*

See [1.1.3]. Ex.1003, ¶292.

[25.1.4] *authenticates the payee; and*

See [1.1.4]. Ex.1003, ¶293.

[25.1.5] *executes ... server, by:*

See [1.1.5]. Ex.1003, ¶294.

[25.1.5.1] *generating ... either:*

See [1.1.5.1]. Ex.1003, ¶295.

[25.1.5.1a]-[25.1.5.1b] *a dynamic ... payee;*

See [1.1.5.1a]-[1.1.5.1b]. Ex.1003, ¶296.

[25.1.5.2] *establishing ... system;*

See [1.1.5.2]. Ex.1003, ¶297.

[25.1.5.3] *outputting ... account; and*

See [1.1.5.3]. Ex.1003, ¶298.

[25.1.5.4] *outputting ... connection;*

See [1.1.5.4]. Ex.1003, ¶299.

[25.2] *receiving ... connection;*

See [1.2]. Ex.1003, ¶300.

[25.3] *storing ... standards;*

See [1.3]. Ex.1003, ¶301.

[25.4] *executing ... API, by:*

See [1.4]. Ex.1003, ¶302.

[25.4.1] *providing ... the payer; and*

See [1.4.1]. Ex.1003, ¶303.

[25.4.2] *processing ... payee.*

See [1.4.2]. Ex.1003, ¶304.

19. 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.*

As explained in limitations [1.1.1]-[1.1.2] and [1.1.5.1], the merchant redirects the buyer's browser to PayPal's website using the URL. Ex.1006, 17 (“PayPal provides a command that [the merchant] use[s] when redirecting [the] buyer's browser to PayPal.” Ex.1006, 17 (Step 3). Ex.1003, ¶¶305-306.

20. Claim 27

[27.1] *The method of claim 25, wherein ... account information.*

See [4.1]. Ex.1003, ¶307.

C. Ground 2: Claims 8-14, and 23-24 are Obvious Under 35 U.S.C. § 103 over Paypal and Schlesser.

1. Summary of Schlesser

Schlesser “relates generally to electronic commerce, and more particularly to secure online transaction processing.” Ex.1020, [0004]. Schlesser discloses a system comprising a customer computer system, a merchant computer system, and a payment processor system. Notably, Schlesser provides evidence that the servers used in financial transactions—such as PayPal's—include hardware such as processors and memory. A more detailed summary of Schlesser can be found in Dr. Mott's declaration. Ex.1003, ¶¶308-311.

2. Reasons to Combine Paypal and Schlesser

A POSITA would have found it obvious for the servers described in PayPal to include hardware such as processors and memory, as evidenced by Schlesser. Ex.1003, ¶¶312-315.

A POSITA would have been specifically motivated to apply Schlesser's merchant computer system and payment processor system to PayPal's methods and systems for conducting an online transaction between a merchant and a buyer using PayPal's secure servers. Ex.1003, ¶313. PayPal explains that "[l]ike many web services, PayPal SOAP is a combination of client-side and server-side schemas, **hardware and software servers**, and core services." Ex.1007, 11. Thus, a POSITA would have been motivated to combine hardware servers or systems with those known in the art for handling online transactions between merchants and buyers. Ex.1003, ¶313.

First, Schlesser describes a payment processor system that includes a central processor and a memory that is "loaded with computer-executable software instructions of a web server." Ex.1020, [0028]. Similar to PayPal, Schlesser's payment processor system "receiv[es] personal account number (PAN) data that is entered into the various fields 60a-d of the payment information form 58 by the customer 12" and "process[es] the payment for completion of the transaction between the customer system 14 and the merchant system 20 using the provided PAN data." Ex.1020, [0036], [0043]. Thus, a POSITA would have been motivated to implement PayPal's servers with the hardware of Schlesser's payment processor system to perform PayPal's method of completing a transaction between a merchant and a buyer. Ex.1003, ¶314. A POSITA would have had a reasonable

expectation of success. Ex.1003, ¶314. Schlesser's payment processor system performs similar processes as PayPal's servers and, thus, a POSITA would have expected that the hardware components of Schlesser's payment processor system could be used in PayPal's servers. Ex.1003, ¶314.

Second, Schlesser's merchant computer system includes a central processing unit and a memory "loaded with computer-executable software instructions of a web server." Ex.1020, [0027]. Similar to PayPal, the merchant computer system establishes a secure link between the customer and the payment processor system so that the customer can provide her PAN data to the payment processor without the merchant computer system receiving the PAN data and sends a transaction processing request to the payment processing system. Ex.1020, [0034], [0029], [0041]. Thus, a POSITA would have been motivated to modify PayPal such that the merchant uses the hardware of Schlesser's merchant computer system to perform PayPal's method of completing a transaction between the merchant and a buyer. Ex.1003, ¶315. A POSITA would have had a reasonable expectation of success. Ex.1003, ¶315. Schlesser's merchant computer system performs similar processes as the merchant's server in PayPal and, thus, a POSITA would have expected that the hardware components of Schlesser's merchant computer system could be used in the merchant's server in PayPal. Ex.1003, ¶315.

3. Claim 8

[8.0] *A secure server that processes a payment transaction from a payer to a payee, the secure server comprising:*

See [1.0]. Ex.1003, ¶316.

[8.1] *one or more processors; and*

A POSITA would have found it obvious that PayPal's secure servers include "one or more processors." Ex.1003, ¶¶317-320. Schlessner teaches a "payment processor system 28 includ[ing] a computer system with **a central processor.**" Ex.1020, [0028]. As discussed at IX.C.2, a POSITA implementing PayPal's method would have found it obvious, and indeed would have been motivated, to use the hardware of Schlessner's payment processing system, including the central processor, in PayPal's servers. Ex.1003, ¶¶317-320.

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

A POSITA would have found it obvious that PayPal's secure servers include "memory storing instructions." Ex.1003, ¶¶321-324. Schlessner explains that a "payment processor system 28 includes...**a memory**" and "is loaded with **computer-executable software instructions** of a web server, as well as those **that implement the methods for securely processing an online transaction.**" Ex.1020, [0028]. As discussed at IX.C.2, a POSITA implementing PayPal's method would have found it obvious, and indeed would have been motivated, to

use the hardware of Schlesser's payment processing system, including the memory with computer-executable instructions, in PayPal's servers. Ex.1003, ¶¶314, 323.

[8.2.1] *provide ... the payer, the API configured to:*

See [1.1]. Ex.1003, ¶325.

[8.2.1.1] *provide ... the payment transaction;*

See [1.1.1]. Ex.1003, ¶326.

[8.2.1.2] *provide ... merchant server;*

See [1.1.2]. Ex.1003, ¶327.

[8.2.1.3] *receive, ... payee;*

See [1.1.3]. Ex.1003, ¶328.

[8.2.1.4] *authenticate the payee;*

See [1.1.4]. Ex.1003, ¶329.

[8.2.1.5] *execute ... server, by:*

See [1.1.5]. Ex.1003, ¶330.

[8.2.1.5.1] *generating ... either:*

See [1.1.5.1]. Ex.1003, ¶331.

[8.2.1.5.1a]-[8.2.1.5.1b] *a dynamic ... the payee;*

See [1.1.5.1a]-[1.1.5.1b]. Ex.1003, ¶332.

[8.2.1.5.2] *establishing ... server;*

See [1.1.5.2]. Ex.1003, ¶333.

[8.2.1.5.3] *outputting ... account; and*

See [1.1.5.3]. Ex.1003, ¶334.

[8.2.1.5.4] *outputting ... connection;*

See [1.1.5.4]. Ex.1003, ¶335.

[8.2.1.6] *receive ... connection;*

See [1.2]. Ex.1003, ¶336.

[8.2.1.7] *store ... standards;*

See [1.3]. Ex.1003, ¶337.

[8.2.1.8] *execute ... the API, by:*

See [1.4]. Ex.1003, ¶338.

[8.2.1.8.1] *providing ... the payer; and*

See [1.4.1]. Ex.1003, ¶339.

[8.2.1.8.2] *processing ... the payee.*

See [1.4.2]. Ex.1003, ¶340.

4. Claim 9

[9.1] *The secure server of claim 8, wherein ... amount;*

See [2.1]. Ex.1003, ¶341.

[9.2-9.3] *process ... the payee.*

See [2.2]. Ex.1003, ¶342-343.

5. Claim 10

[10.1] *The secure server of claim 9, wherein ... data token.*

See [3.1]. Ex.1003, ¶344.

6. Claim 11

[11.1] *The secure server of claim 8, wherein ... account information.*

See [4.1]. Ex.1003, ¶345.

7. Claim 12

[12.1] *The secure server of claim 8, wherein ... the payer.*

See [5.1]. Ex.1003, ¶346.

8. Claim 13

[13.1] *The secure server of claim 8, wherein ... the payer.*

See [6.1]. Ex.1003, ¶347.

9. Claim 14

[14.1] *The secure server of claim 8, wherein ... account number.*

See [6.1]. Ex.1003, ¶348.

10. Claim 23

[23.0] *A merchant server, operated by or for the benefit of a payee, comprising:*

See [15.0]. Ex.1003, ¶349.

[23.1] *one or more computer processors; and*

A POSITA would have found it obvious for a merchant server to include
“*one or more computer processors.*” Ex.1003, ¶¶350-353. Schlessner teaches a

“merchant system 20...includes **a central processor.**” Ex.1020, [0027]. As discussed at IX.C.2, a POSITA implementing PayPal’s method would have found it obvious, and indeed would have been motivated, to use the hardware of Schlesser’s merchant computer system, including the central processor, as the merchant server. Ex.1003, ¶315, 351.

[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:*

A POSITA would have found it obvious for a merchant server to include “*non-transitory computer readable storage media storing instructions.*” Ex.1003, ¶354-356. Schlesser teaches a “merchant system 20...includes...**a memory**” and is “**loaded with computer-executable software instructions** of a web server and those **that implement one or more of the disclosed methods for securely processing an online transaction.**” Ex.1020, [0027]. As discussed at IX.C.2, a POSITA implementing PayPal’s method would have found it obvious, and indeed would have been motivated, to use the hardware of Schlesser’s merchant computer system, including the memory with computer-executable instructions, in the merchant’s servers. Ex.1003, ¶315, 354. A POSITA would have understood that a memory is a “*non-transitory computer readable storage media.*” Ex.1003, ¶355.

[23.2.1] *provide ... a payer;*

See [15.1]. Ex.1003, ¶357.

[23.2.2] *receive ... payer;*

See [15.2]. Ex.1003, ¶358.

[23.2.3] *provide, ... the payee;*

See [15.3]. Ex.1003, ¶359.

[23.2.4] *display ... computing system;*

See [15.4]. Ex.1003, ¶360.

[23.2.5] *provide ... the payee;*

See [15.5]. Ex.1003, ¶361.

[23.2.6] *initiate ... function:*

See [15.6]. Ex.1003, ¶362.

[23.2.6.1] *establishing ... or frame;*

See [15.6.1]. Ex.1003, ¶363.

[23.2.6.2] *outputting ... account; and*

See [15.6.2]. Ex.1003, ¶364.

[23.2.6.3] *outputting ... connection;*

See [15.6.3]. Ex.1003, ¶365.

[23.2.7] *initiate ... merchant server;*

See [15.7]. Ex.1003, ¶366.

[23.2.8] *receive ... the payer;*

See [15.8]. Ex.1003, ¶367.

[23.2.9] *store ... the payer; and*

See [15.9]. Ex.1003, ¶368.

[23.2.10] *process ... amount.*

See [15.10]. Ex.1003, ¶369.

11. Claim 24

[24.1] *The merchant server of claim 23, wherein ... information.*

See [3.1]. Ex.1003, ¶370.

X. CONCLUSION

Petitioner has established a reasonable likelihood that the Challenged Claims are unpatentable.

Respectfully submitted,

Dated: July 18, 2024
HAYNES AND BOONE, LLP
2801 N. Harwood Street
Suite 2300
Dallas, TX 75201
Customer No. 27683

/Calmann J. Clements/
Calmann J. Clements
Lead Counsel for Petitioner
Registration No. 66,910

XI. MANDATORY NOTICES

A. Real Party-in-Interest

Pursuant to 37 C.F.R. § 42.8(b)(1), Petitioner certifies that the real parties-in-interest are M&A Ventures, LLC, Repay Holdings, LLC, and Repay Holdings Corporation.

B. Related Matters

Pursuant to 37 C.F.R. § 42.8(b)(2), to the best knowledge of the Petitioner, the '621 patent is or was involved in the following cases:

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

C. Lead and Back-up Counsel and Service Information

Lead Counsel

Calmann J. Clements
HAYNES AND BOONE, LLP
2801 N. Harwood Street
Suite 2300
Dallas, TX 75201

Phone: (972) 739-8638
Fax: (214) 200-0853
calmann.clements.ipr@haynesboone.com
USPTO Reg. No. 66,910

Backup Counsel

Andrew S. Ehmke
HAYNES AND BOONE, LLP
2801 N. Harwood Street
Suite 2300
Dallas, TX 75201

Phone: (214) 651-5116
Fax: (214) 200-0853
andy.ehmke.ipr@haynesboone.com
USPTO Reg. No. 50,271

Brooke E. Parker
HAYNES AND BOONE, LLP
2801 N. Harwood Street
Suite 2300
Dallas, TX 75201

Phone: (972) 739-8648
Fax: (214) 200-0853
brooke.parker.ipr@haynesboone.com
USPTO Reg. No. 79,943

Please address all correspondence in this proceeding to lead and back-up counsel. Petitioner consents to service in this proceeding by email at the addresses above.

XII. CLAIM APPENDIX

Claim 1

[1.0] 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,

[1.1.5.1a] the URL comprising either: a dynamic URL generated by the secure server for the payer and the payee;

[1.1.5.1b] the URL comprising either: ... 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;

[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.

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;

[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.

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.

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.

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.

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.

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.

Claim 8

[8.0] 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;

[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

[8.2.1.5.1b] 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:

[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.

Claim 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

[9.3] forward at least a portion of the second payment amount to the payee.

Claim 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.

Claim 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.

Claim 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.

Claim 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 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.

Claim 15

[15.0] 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;

[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.

Claim 16

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

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.

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.

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).

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.

Claim 21

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

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.

Claim 23

[23.0] 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

[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.

Claim 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.

Claim 25

[25.0] 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:

[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;

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

[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:

[25.1.5.1a] a dynamic URL generated by the secure server for the payer and the payee; or

[25.1.5.1b] 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;

[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.

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.

Claim 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.

CERTIFICATE OF WORD COUNT

Pursuant to 37 C.F.R. § 42.24(d), Petitioner hereby certifies, in accordance with and reliance on the word count provided by the word-processing system used to prepare this Petition, that the number of words in this paper is 13,909. Pursuant to 37 C.F.R. § 42.24(d), this word count excludes the table of contents, table of authorities, mandatory notices under § 42.8, certificate of service, certificate of word count, appendix of exhibits, and any claim listing.

Dated: July 18, 2024

/Calmann J. Clements/
Calmann J. Clements
Lead Counsel for Petitioner
Registration No. 66,910

CERTIFICATE OF SERVICE

The undersigned certifies that, in accordance with 37 C.F.R. § 42.6(e) and 37 C.F.R. § 42.105, service was made on Patent Owner as detailed below.

Date of service July 18, 2024

Manner of service FEDERAL EXPRESS

Documents served

- Petition for IPR of U.S. Patent No. 11,620,621
- Certificate of Word Count
- Petitioner's Exhibit List
- Exhibits Ex.1001 to Ex.1025

Persons served Blank Rome LLP
1825 Eye Street NW
Washington, DC 20006-5403

/Calmann J. Clements/
Calmann J. Clements
Lead Counsel for Petitioner
Registration No. 66,910