Entersekt

Improving checkout with SPC.

GERHARD OOSTHUIZEN
ERHARD BRAND
ARNO VAN DER MERWE

MARCH 2021

Agenda and purpose

Improving eCommerce checkout

Background

- eCommerce checkout challenges
- Various technologies/standards aiming to improve checkout
- Industry requirements for a solution

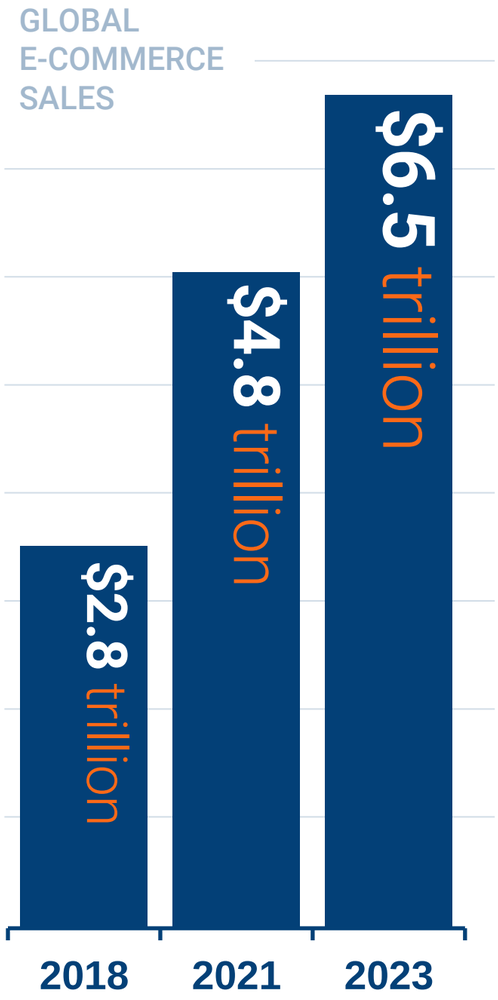
Proposal

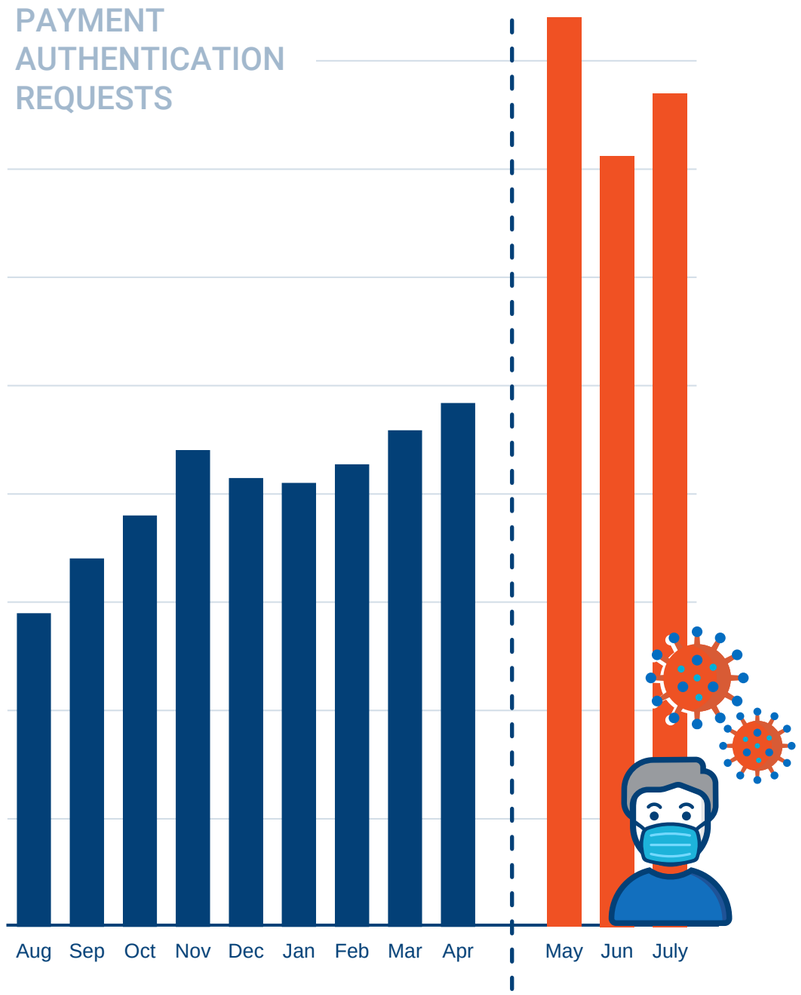
A. Expanding on SPC to offer a lower friction challenge
B. Removing friction; creating a silent browser identifier for Risk based Auth

eCommerce **challenges!**

Abandonment and false declines

Trends we're seeing **in the market.**





User experience is what will make the difference.

65% of customers abandoned carts due to friction

$100s OF BILLIONS in sales foregone due to friction at checkout

AND THAT'S ALL WITHOUT TAKING FRAUD INTO ACCOUNT

SOURCE: https://content.ekata.com/Consumers-Demand-Speed-and-Security-in-the-Digital-Experience.html

The price is high for **poor checkout experiences.**

Bad checkout experiences result in abandonment

Detection and prevention tools can have a negative impact

$146B

in card-not-present purchases are declined per year

52%

of orders declined for fraud were good orders to fulfill

62%

of cardholders will abandon a declined card

Sources: Ethoca Research, Solving the CNP False Decline Puzzle, Visa

Notes from Microsoft (as a Merchant)

Bad checkout experiences continue to hurt customers and lose merchants revenue…

- Authentication success rates are still too low
  - Browser-based is 75%, app-based is 45%.
- Abandonment is too high
  - Browser-based is 13%, app-based is 18%.
- Challenge rates are much too high.
  - Browser-based is 81%, app-based is 75%.

“…The payments ecosystem must find ways to lower the challenge rate…”, whether leveraging exemptions under EMV 3DS v2.2, refining risks models or exploring delegated authentication use cases

Approval rates **improve** when a **challenge succeeds**

https://www.linkedin.com/posts/deanjordaan_sca-psd2-3ds-activity-6763544640764411904-nCuk

Improvements in the works.

Various technologies/standards aiming to improve checkout

What is 3-D Secure?

3-D Secure is a messaging protocol enabling issuers to authenticate consumers during online shopping

- 3D Secure 2.2 is the latest version of this protocol, currently being rolled out worldwide
- The Challenge flow (same as 3D Secure 1.0) executes inside an iFrame
  - For 3D Secure 2 this should be the exception (not more than 20% of transactions)
- Frictionless flow (using Risk based Authentication)
  - This utilizes a hidden iframe (called MethodURL) to capture browser context/information

The emergence of **FIDO and WebAuthn.**

- Original focus on Web Login
- WebAuthn offers PSD2 SCA compliant authentication from within a browser
- Supported by various parties, such as all the major OS and browser providers (Microsoft, Google, Apple, and of course Entersekt…

USER VALIDATION
(USER PRESENCE)
Is there a person there?

USER VERIFICATION
(MULTIFACTOR AUTH)
Is the right person there?

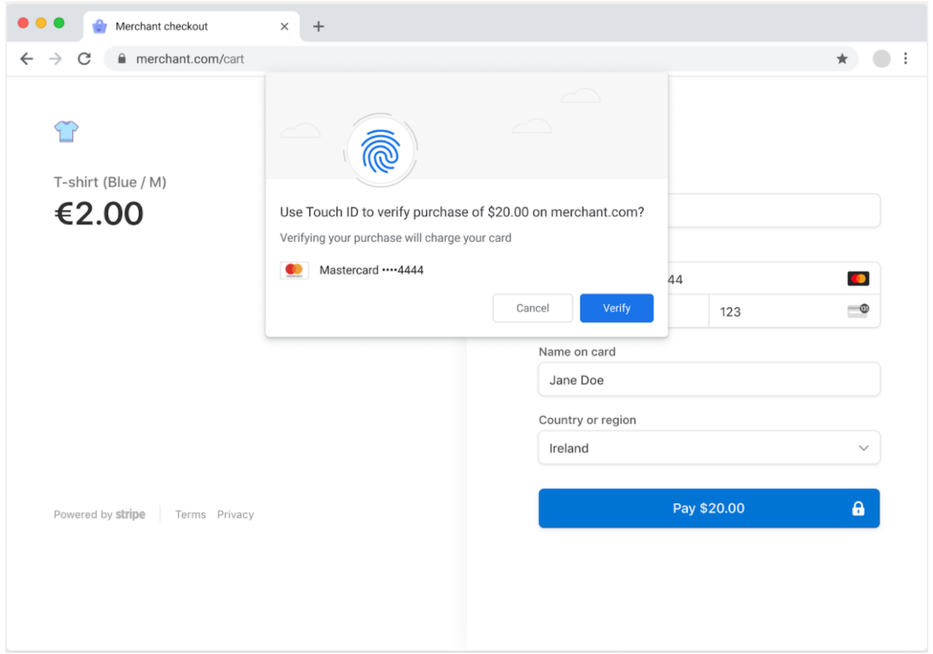
ROAMING AUTHENTICATORS
Implemented off device

PLATFORM AUTHENTICATORS
Built into a device platform

Secure Payment Confirmation

https://rsolomakhin.github.io/pr/spc/

- This is a great move forward for the web!
- Cross-domain **predictability**
  - Merchant controls the experience
  - Issuer (Bank) controls the identity
- **Payment** focused **display**
  - Better customer experience
  - Closer to regulatory intent (PSD2)
- SPC requires **both parties to support** it
  - 3D Secure gives issuers control of UI
  - If an issuer implements this, they will still use it in their challenge flows!



Can we create a bridge between these worlds?

No challenge
may lead to
higher false decline rates
(eroding customer confidence)

Merchant Controls UX
Issuer controls ID

Secure payment confirmation (SPC) with WebAuthn

Can we add to this?

Further reduce the clicks/steps

SPC with one click

Silent challenge

Increased approval rates

Increased friction
(additional clicks/input and unpredictable UX)
may lead to
higher abandonment rates

Industry **Requirements.**

Solutions must meet these needs

PSD2 and SCA **background.**

- PSD2 regulation introduced the need for Strong Customer Authentication (SCA) on all remote transactions, unless exempted.
- This delivers a dramatic improvement in security but may negatively affect the user experience.
- SCA only required in 20% of transactions
- Various other exceptions and rules come into play

PSD2 SCA requires the use of at least two of the following three factors:

Authentication outside of Europe

SOMETHING YOU
KNOW

SOMETHING YOU
HAVE

SOMETHING YOU
ARE

- A single possession (“something you have”) factor is typically used for payment consent (3D Secure) outside of Europe
  - E.g. SMS OTP, App based OTP, Out of Band Push Authentication to Mobile
- Provable possession is a very strong signal for Risk Based Authentication
  - A core driver behind the browser fingerprinting used in EMV 3D Secure
  - Typically, an additional challenges is not needed if the browser is known

Requirements for a solution

- It would have to be **privacy** friendly…
  - Domain bound (so only visible to the party that issued it)
  - Accessible using an identifier only known by the issuer thereof (on the server)
  - Get user consent before it’s issued to that user
  - Allow the user to clear their history and remove this possession factor.
- … and **secure**
  - A secret generated in the browser, that can never be copied
  - Cryptographic proof for every interaction based on a server challenge

We can learn from current W3C standards

- WebAuthn has been designed with privacy & security in mind
  - *Big differences are hardware backed storage and physical user action*
- Secure Payment Confirmation enables control to merchants and a lower friction experience for consumers
- Credential Management enables the storage of secrets such as passwords and public key credentials
- Web Crypto can generate a public-private keypair with a protected private key, and enables signing a challenge

With this foundation, we can create a possession factor
And use it to sign a payment transaction

Why would the alternatives not work

- Using Server-side Cookies
  - No User Consent
  - Nothing is signed (no replay protection)
  - Always provided for full domain (not linked to specific credential ID)
- WebID
  - This is an OAuth2 based API preventing direct comms between RP & IDP
  - Payment use-cases not currently covered
  - The IDP still needs to complete a user-challenge; WebID does not solve for this
- Trust Tokens
  - Mechanism to enable anonymous user grouping. Not really what is needed to trust a user on a specific browser.
  - No consent required

Further **friction** reduction.

Utilizing the browser as a possession factor, with single-click user consent

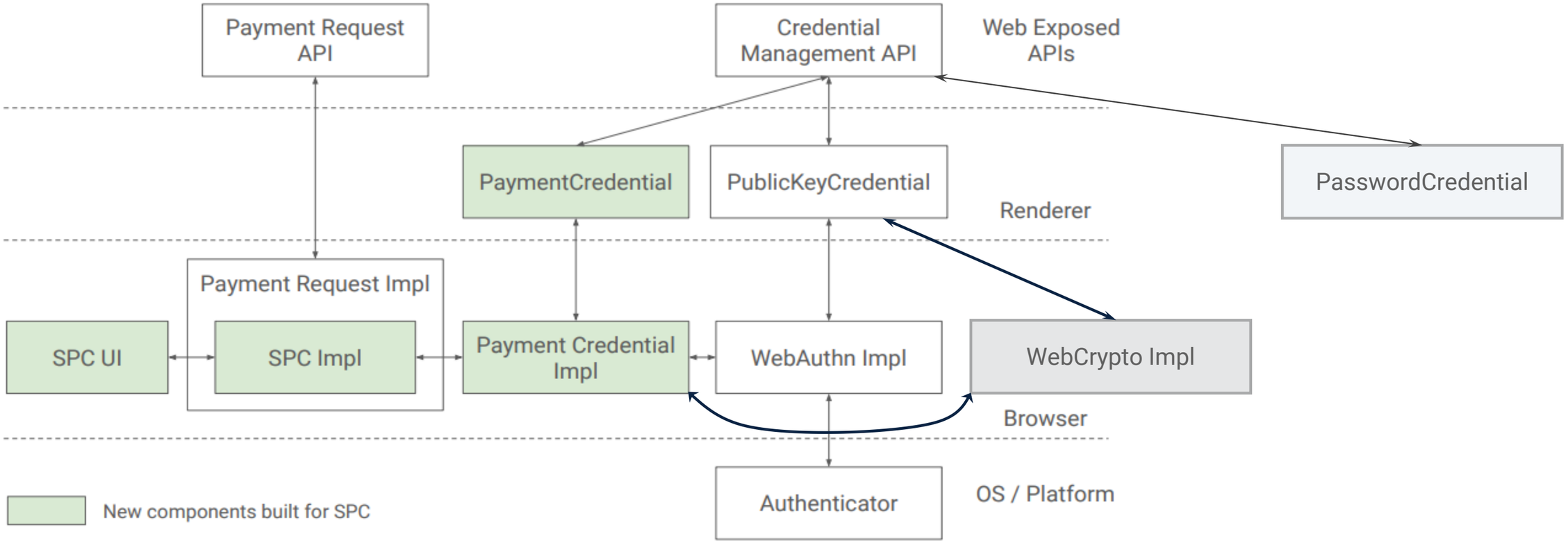
Proposal for a new possession-only factor

- Enable the browser agent to issue a possession factor
  - User specific (Credential ID) and bound to a specific domain…
  - Only generated after explicit consent
- Stored within Credential Management

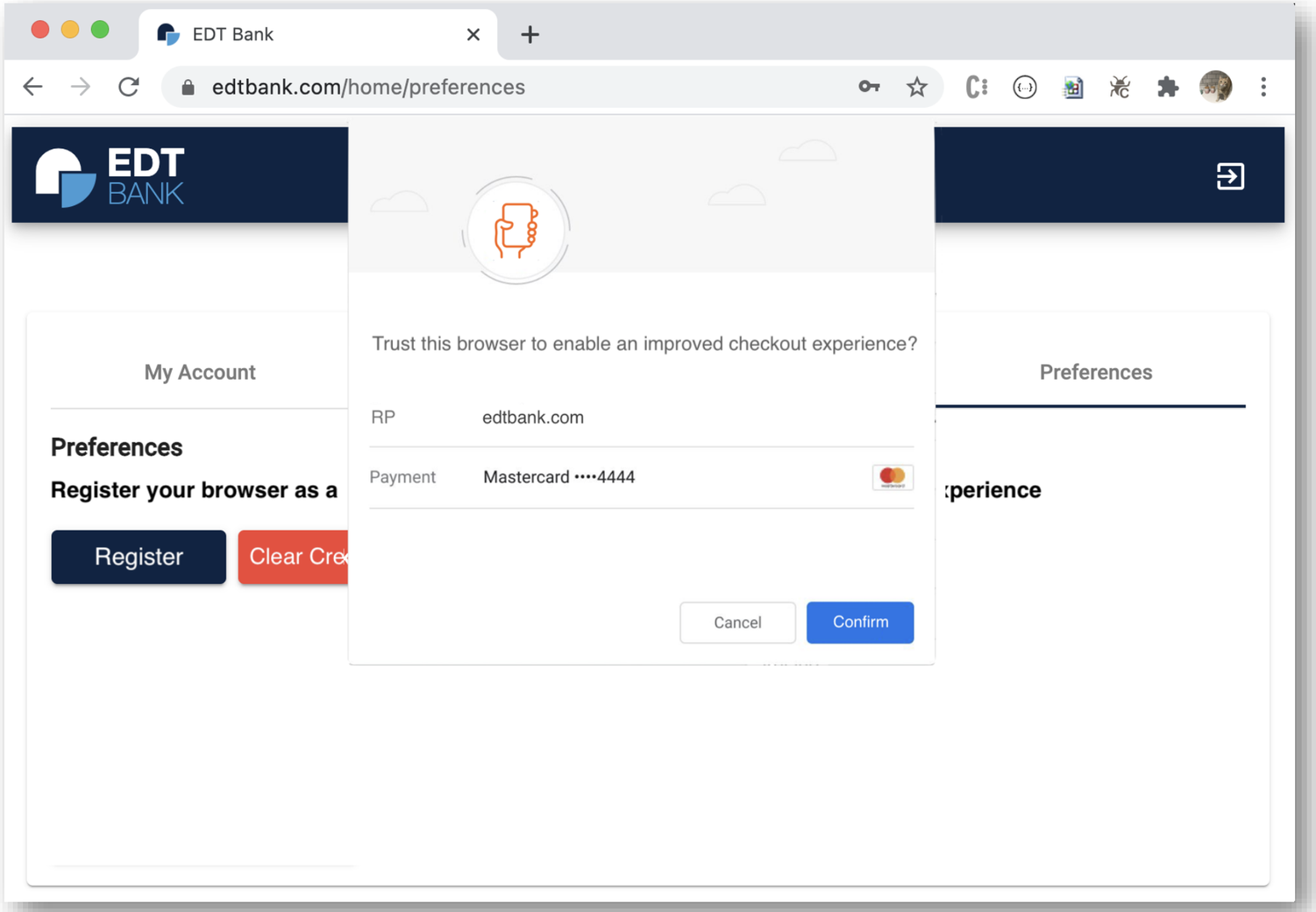
- Link this possession credential to a payment credential, binding the browser and payment mechanism (e.g. Card)

Architectural reflections …

Extended from the Chrome team's proposal



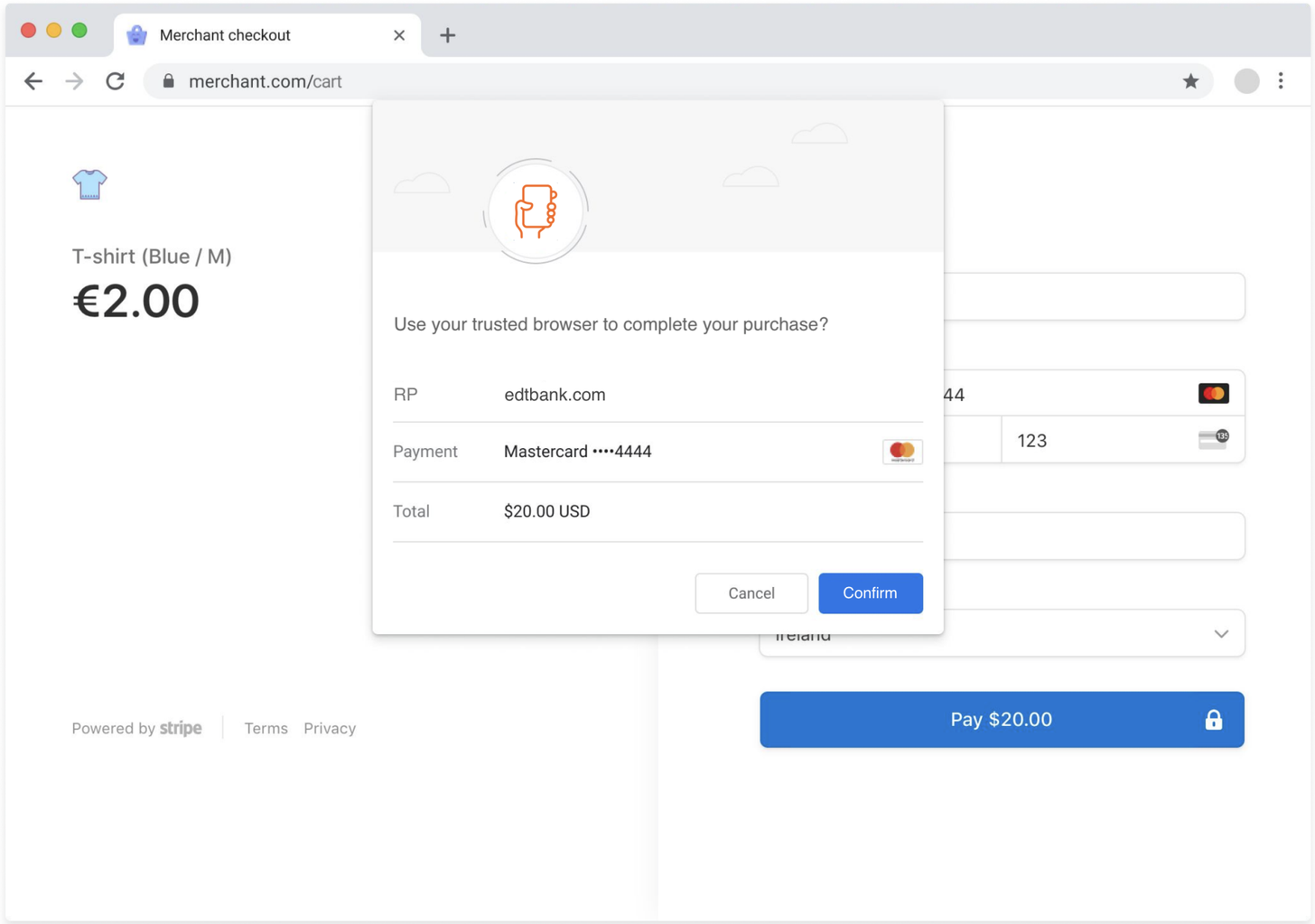
Creation consent



The credential will **only** be **issued after** explicit **user consent**.

Enable the consumer to **manage** (including delete) **the credential** at any stage (similar to passwords), from their browser agent.

Authentication consent with only 1 click

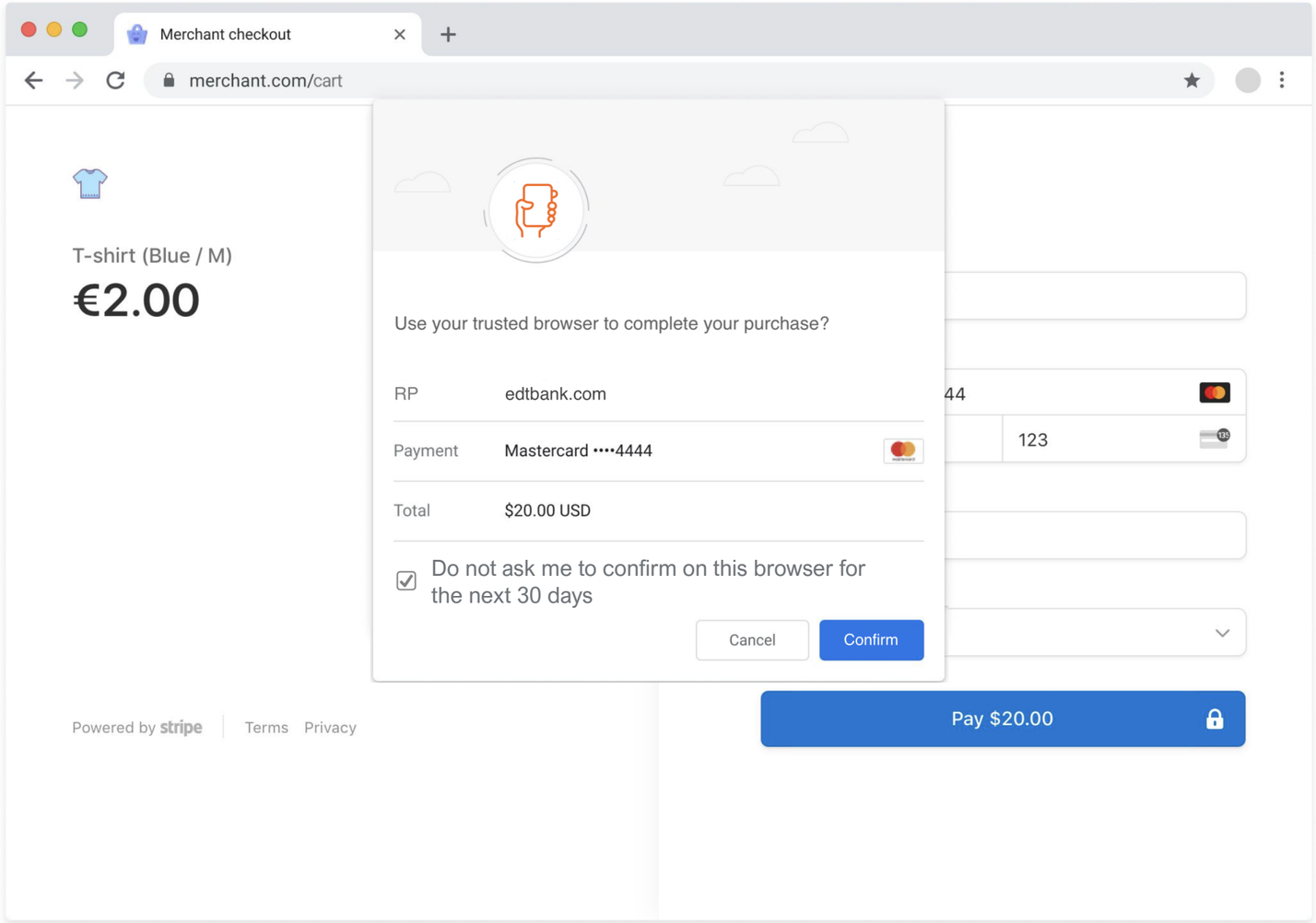


For authentication, a **challenge** will only be **signed after** a **user action**.

Allowing the Relying party (bank/issuer) to choose the required level of trust

- Full SCA (WebAuthn)
- Possession only

Consent for a frictionless flow



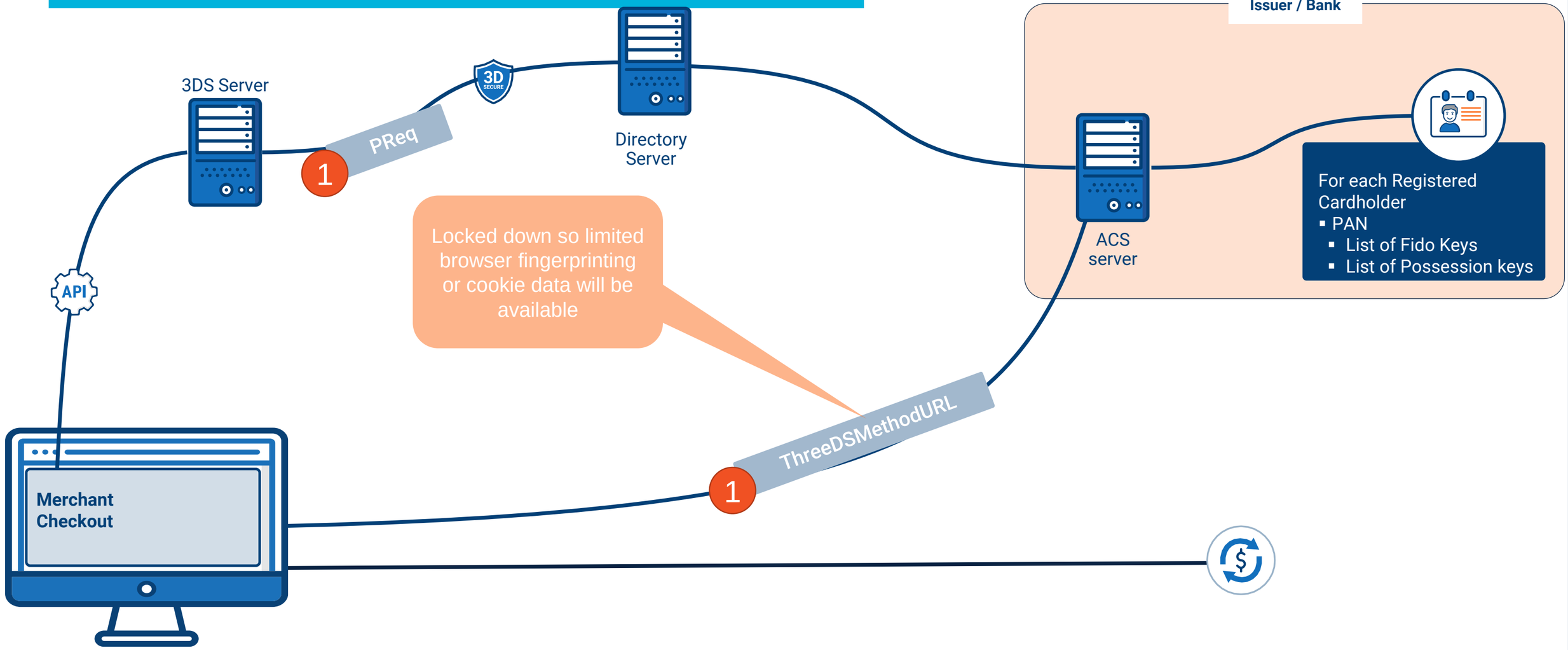
Potential enhancements

- Enable customer to skip the confirmation for a certain period.
- Perhaps consent for a frictionless challenge should only be granted based on a SCA / Full WebAuthn consent

- Should the **issuer** be able to also indicate if a user challenge is not needed (enabling a frictionless flow) without customer consent?

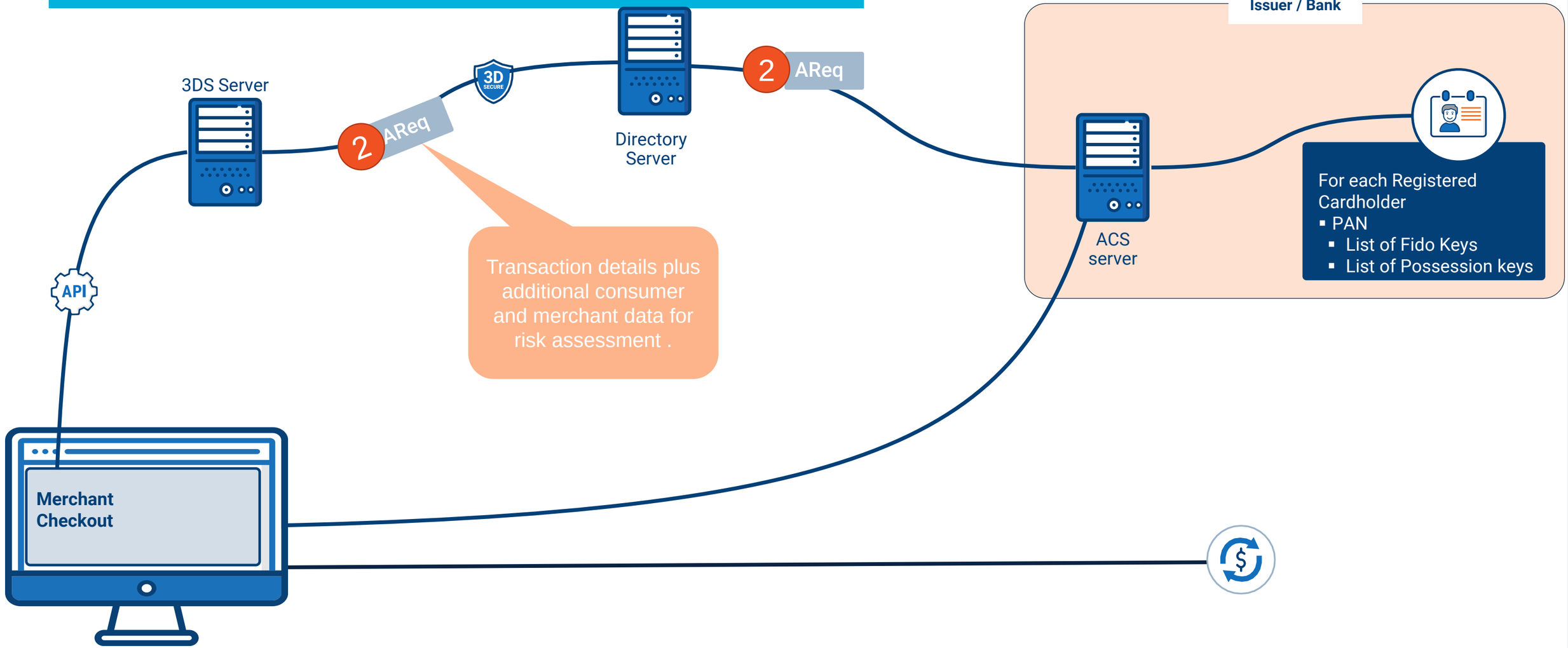
Integration **view: Authentication.**

Frictionless challenge…



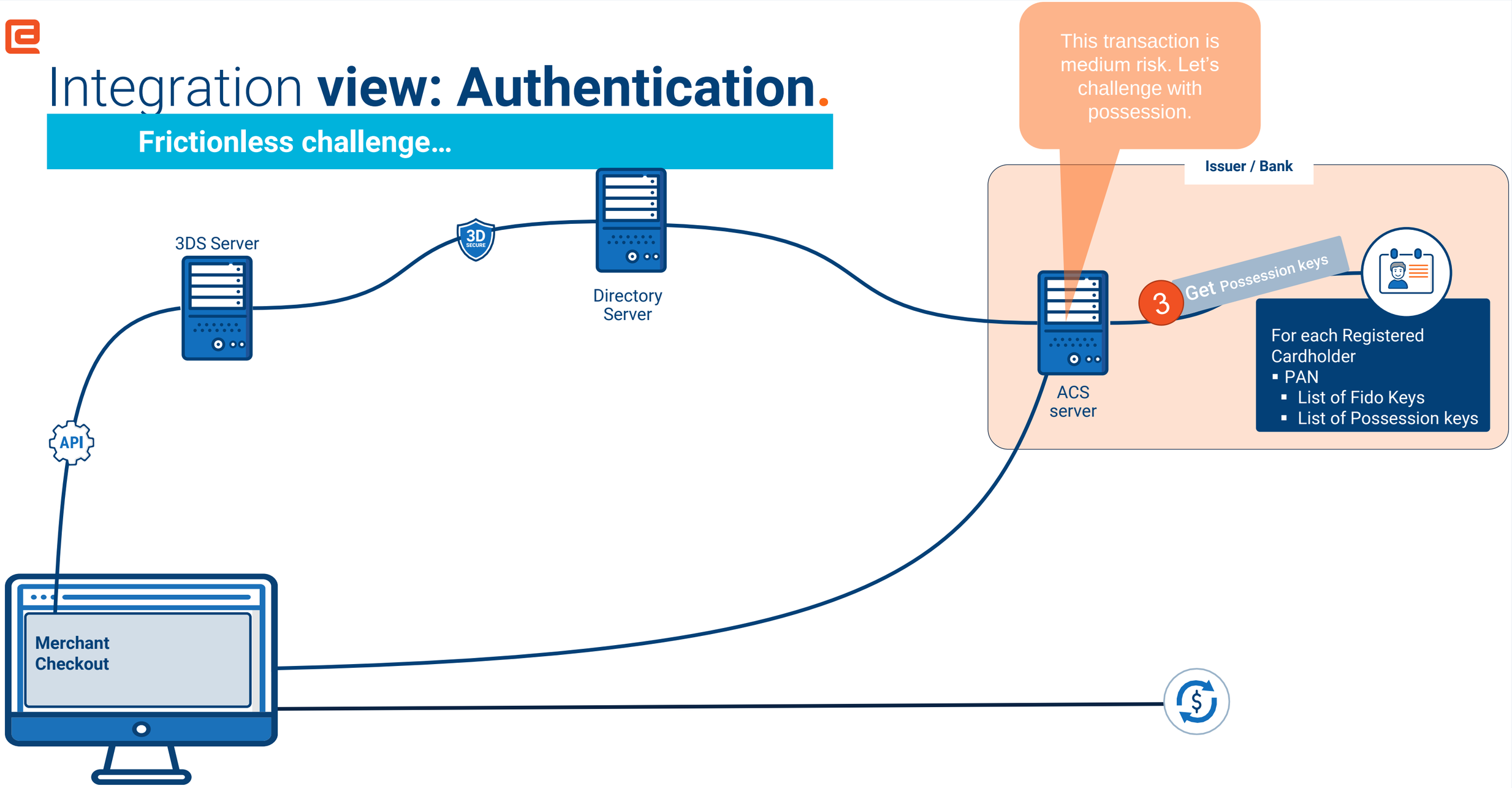
Integration **view: Authentication.**

Frictionless challenge…



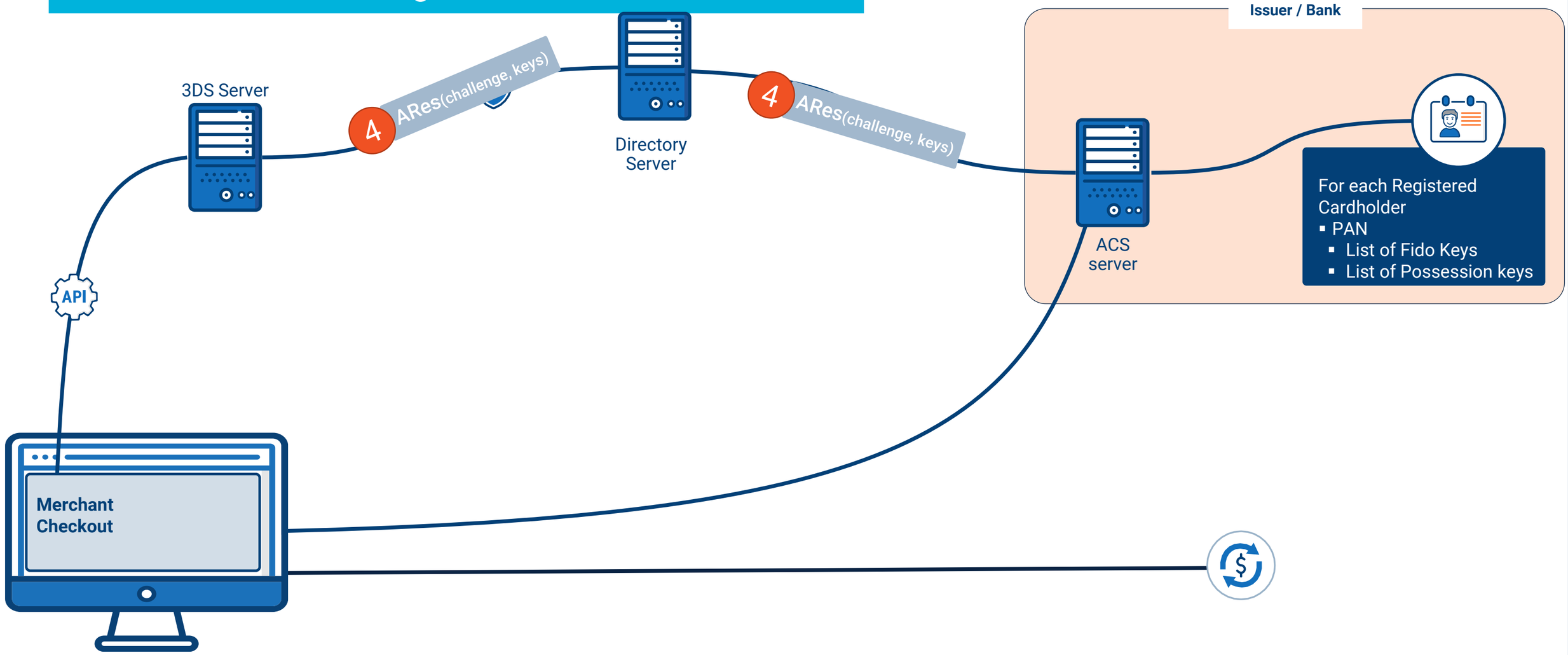
Integration **view: Authentication.**

Frictionless challenge…



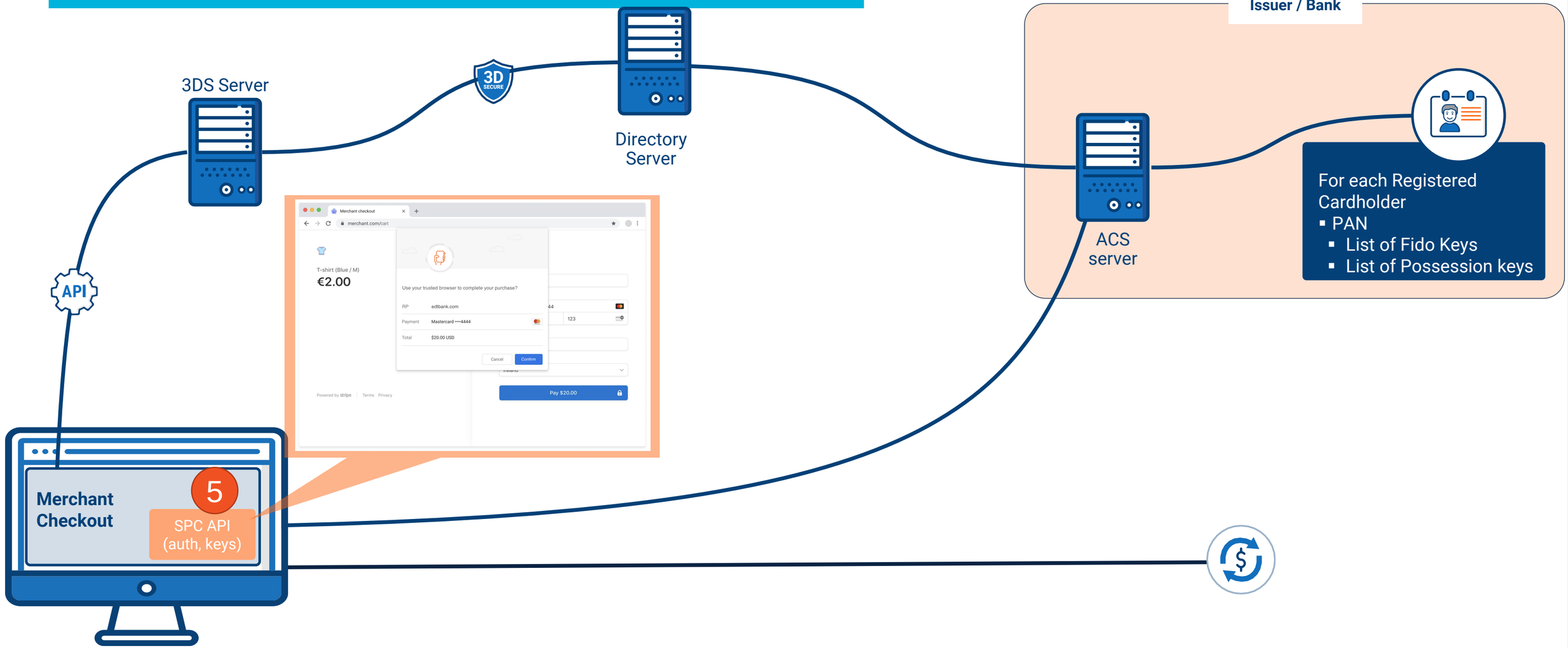
Integration **view: Authentication.**

Frictionless challenge…



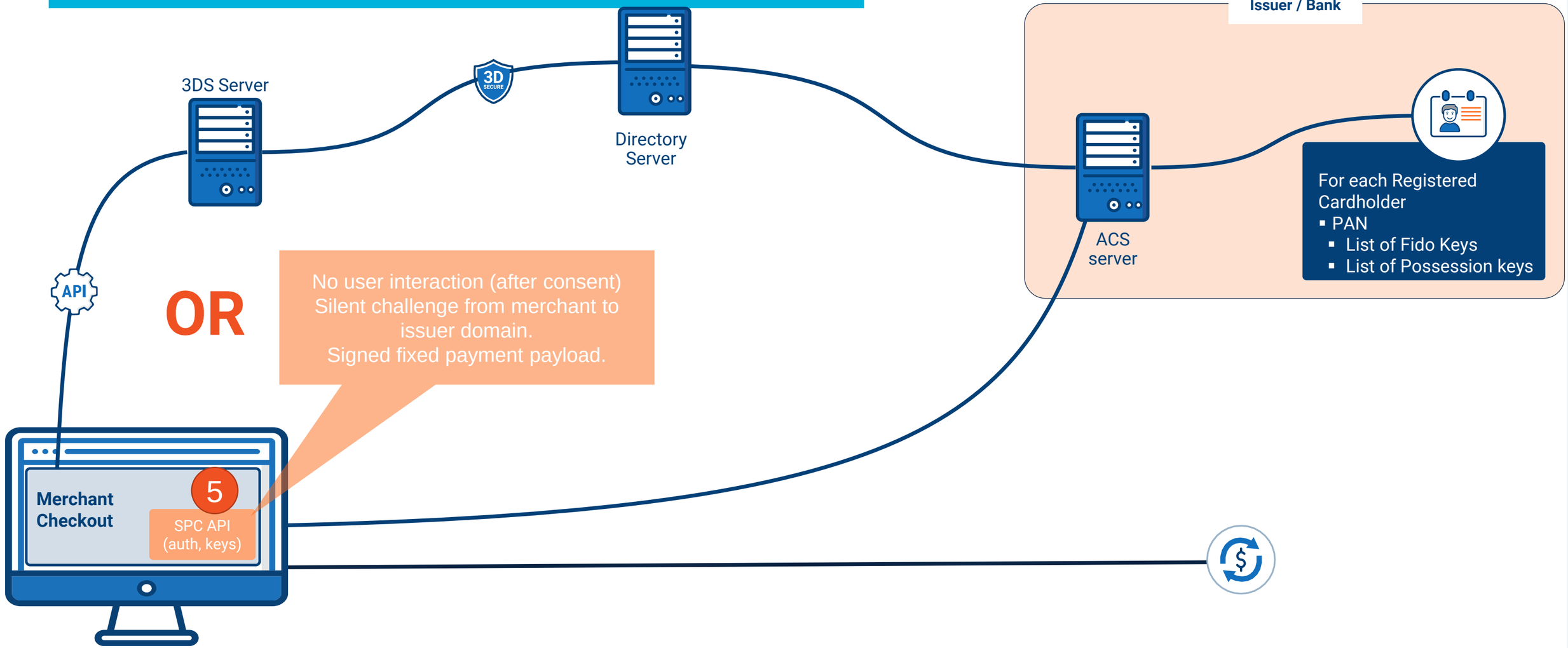
Integration **view: Authentication.**

Frictionless challenge…



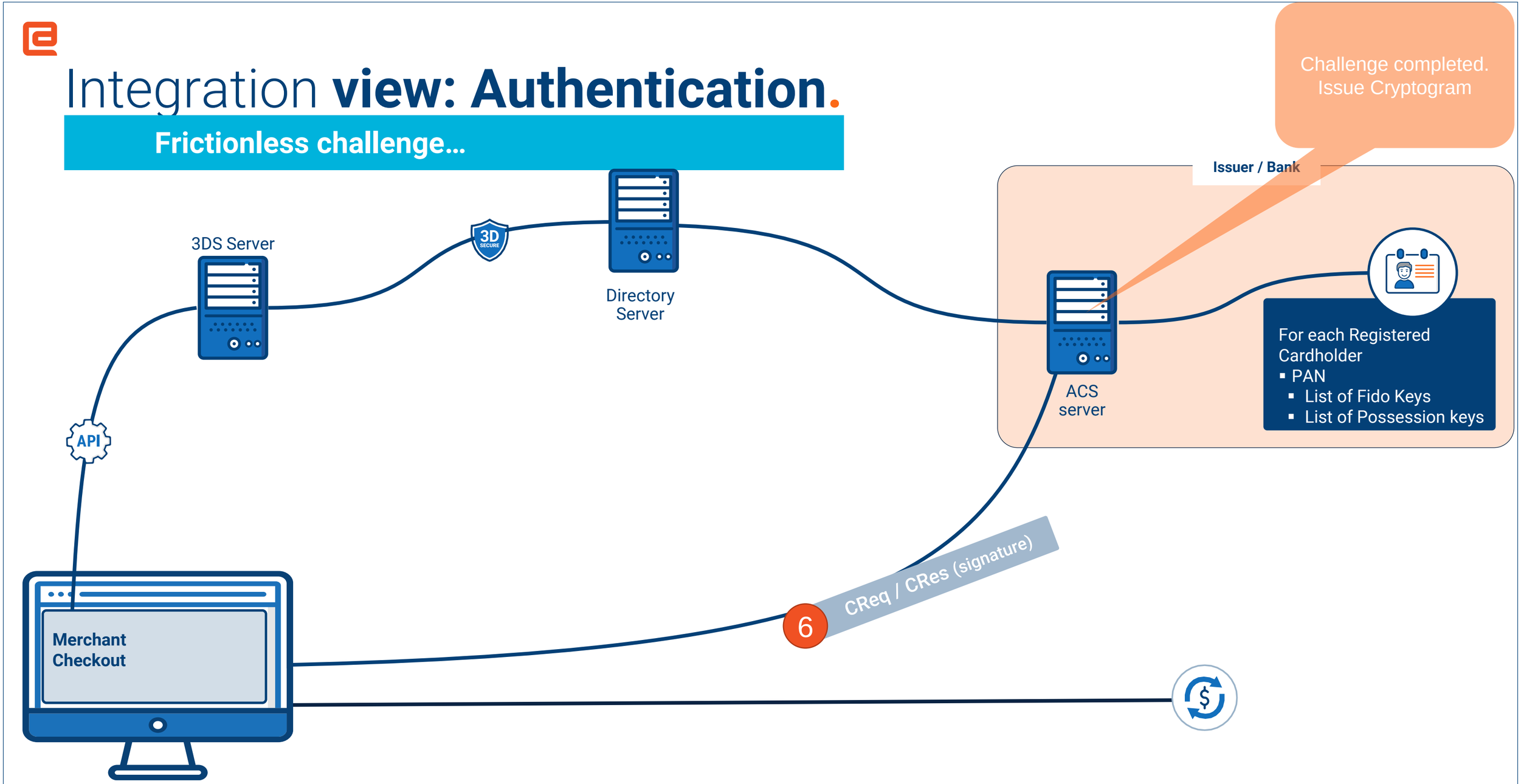
Integration **view: Authentication.**

Frictionless challenge…



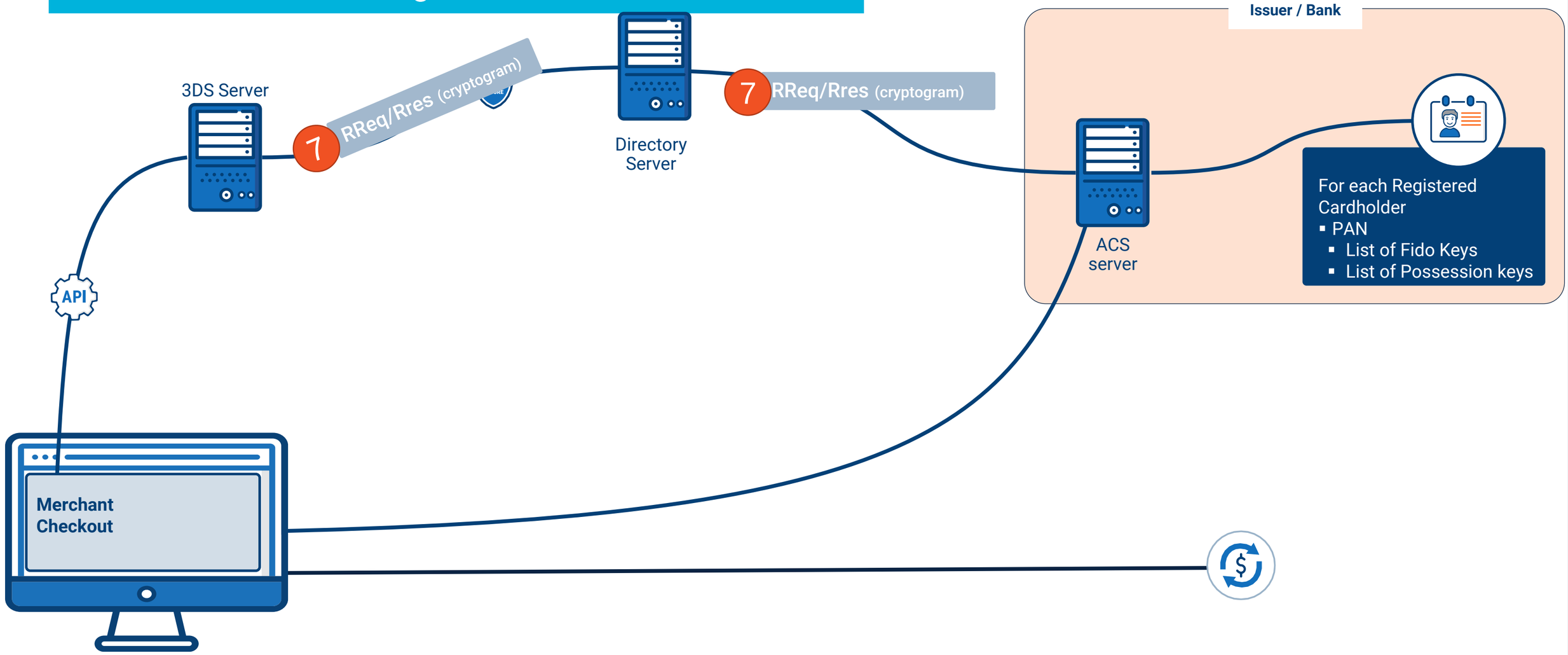
Integration **view: Authentication.**

Frictionless challenge…



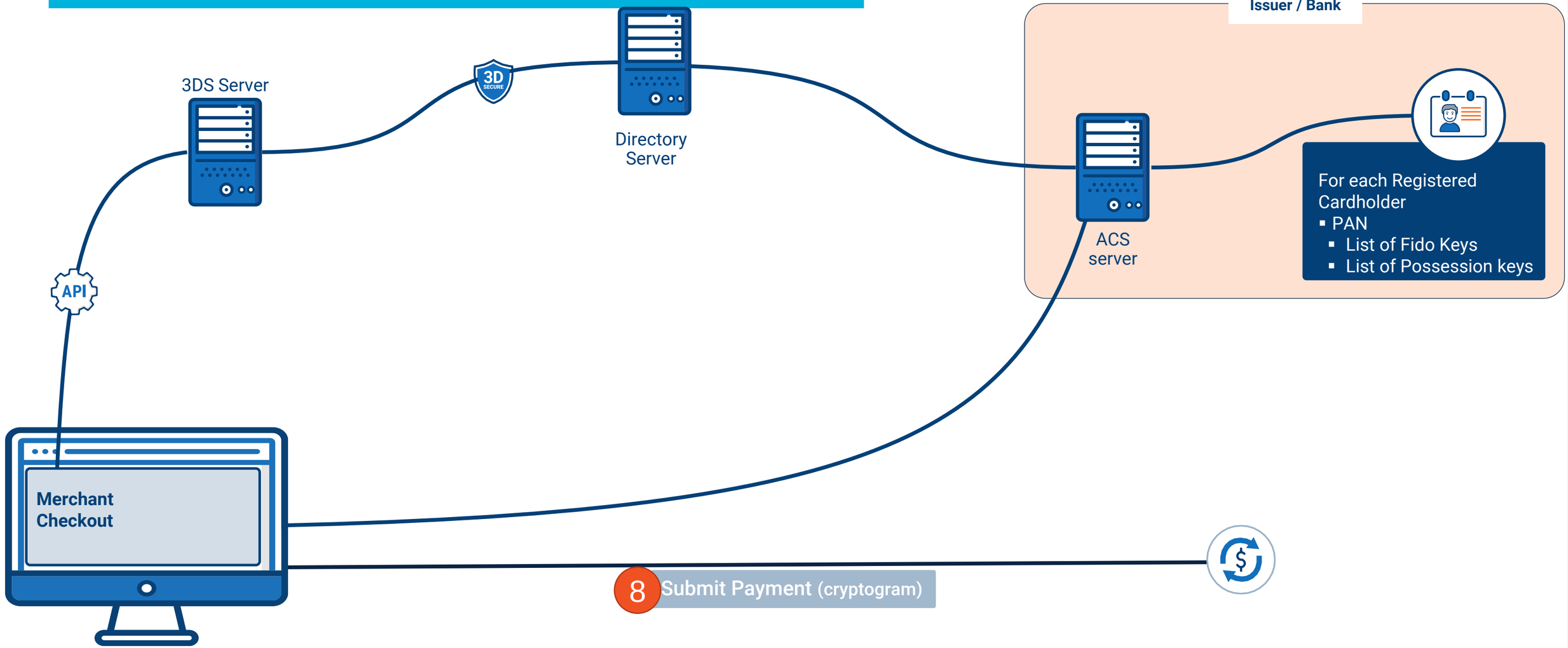
Integration **view: Authentication.**

Frictionless challenge…



Integration **view: Authentication.**

Frictionless challenge…



Impact on 3D Secure flow

- The technique aligns fully with the current SPC proposal
  - It can work with 3D Secure 2.1 and later
  - The merchant would not have to be aware of the difference between the two pairs of keys, but practically this could add value
- The proposed SPC solution does require merchant integration
  - Although as stated, most issuers will also implement this from their domain as part of their challenge flow (inside their challenge windows

As with SPC, this proposal is not specific to 3D Secure.
It is generic to supports other payment instruments and rails.

Removing **friction.**

Utilizing the browser as a silent possession factor

EMV 3D Secure (v2) browser requirements

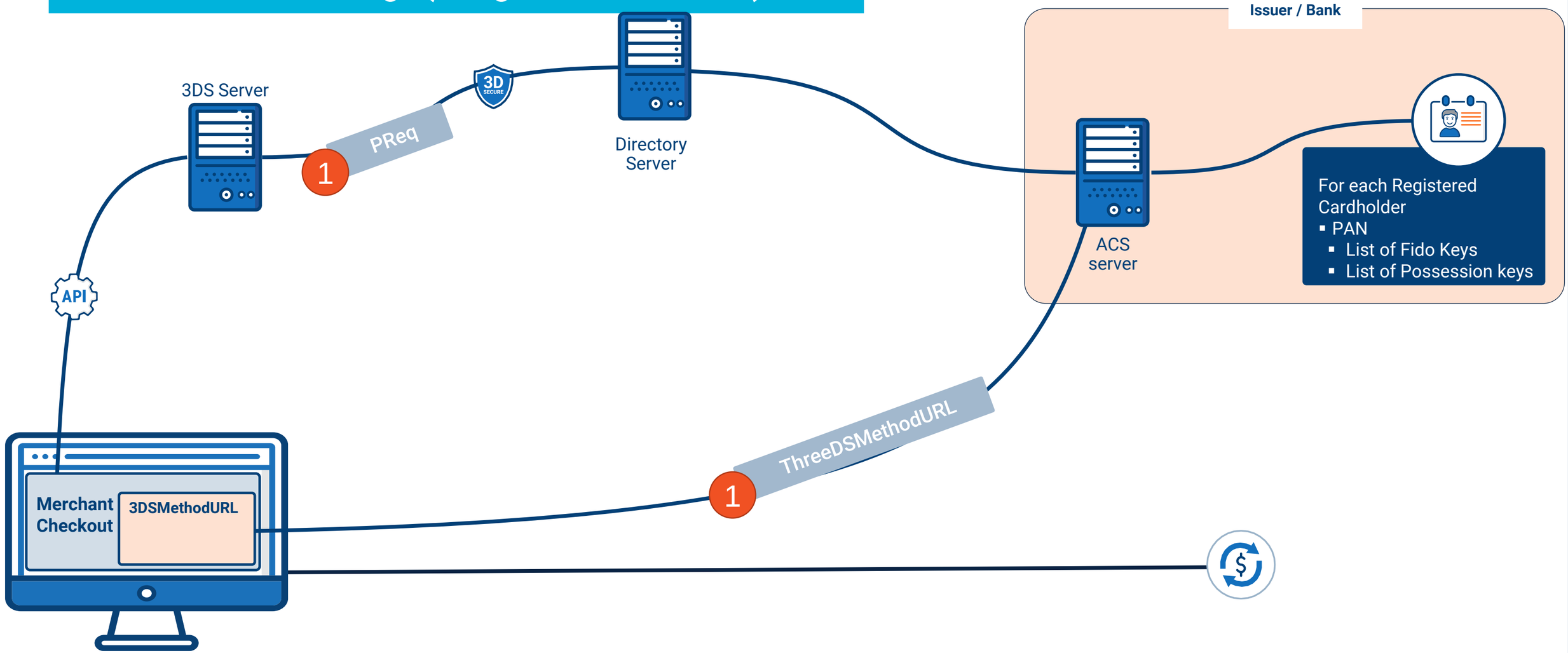
- 3D Secure 2 design caters for frictionless authentication
  - Added largely due to address merchant frustrations with 3DS 1 and low success rates
  - To protect the user, this requires Risk Based Authentication (RBA)
  - The mechanism (3DSMethodURL) was intended to perform browser fingerprinting, which is not desirable anymore (https://www.w3.org/TR/fingerprinting-guidance/)
- The Risk Based Auth (RBA) logic aims to leverage 3 data sources
  - Browser/Device Data: To identify a familiar or trusted device
  - User Data: To link the user with a history of using this device
  - Transaction Data: To correlate if this user/device engages in this type of transaction
- *A potential solution should cater for all these data points*

A unique browser id will also help here!

- EMVCo indicated a need for a better browser identifier
  - The current ‘risk’ method is based on customer data being captured an analyzed
  - They want to protect privacy, so are open to an alternative
  - https://www.w3.org/2021/02/emvco-wpsig.pdf from WPSIG call (4 Feb 2021)
- We could use the same possession factor concept
  - Enable its use silently inside iFrames
  - Issuing of credential would still require user consent
- The challenge could bind **browser + user + transaction** details
  - Enabling Risk based authentication without the need for browser fingerprinting

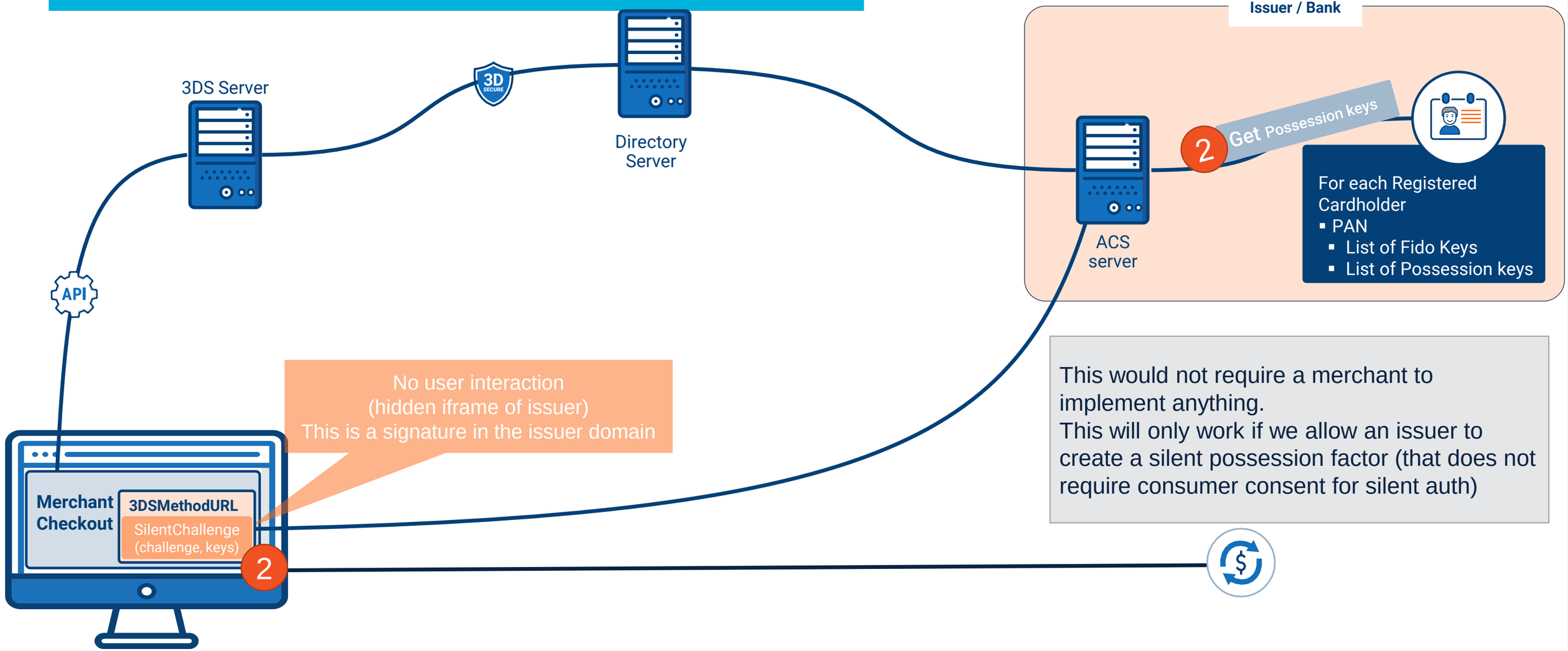
Integration **view: Authentication.**

Without a challenge (using 3DS Method URL)



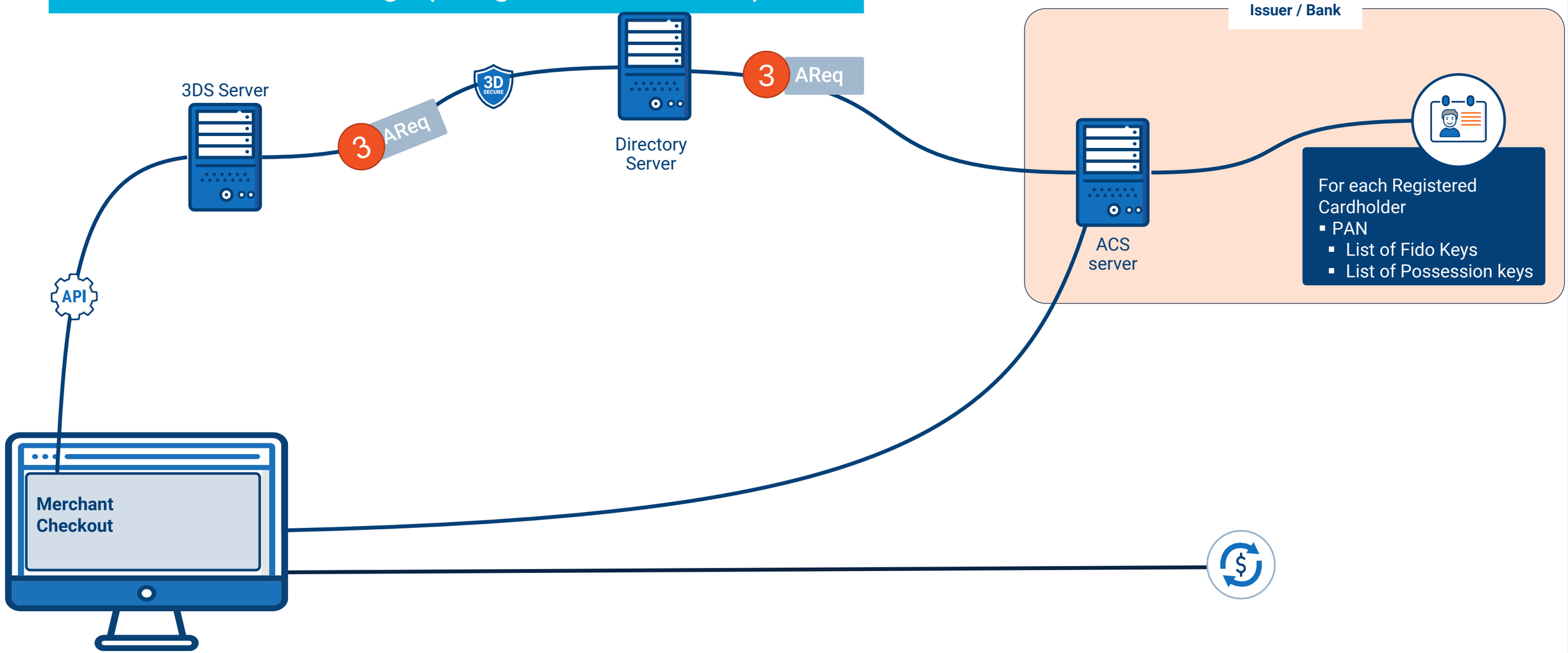
Integration **view: Authentication.**

Without a challenge (using 3DS Method URL)



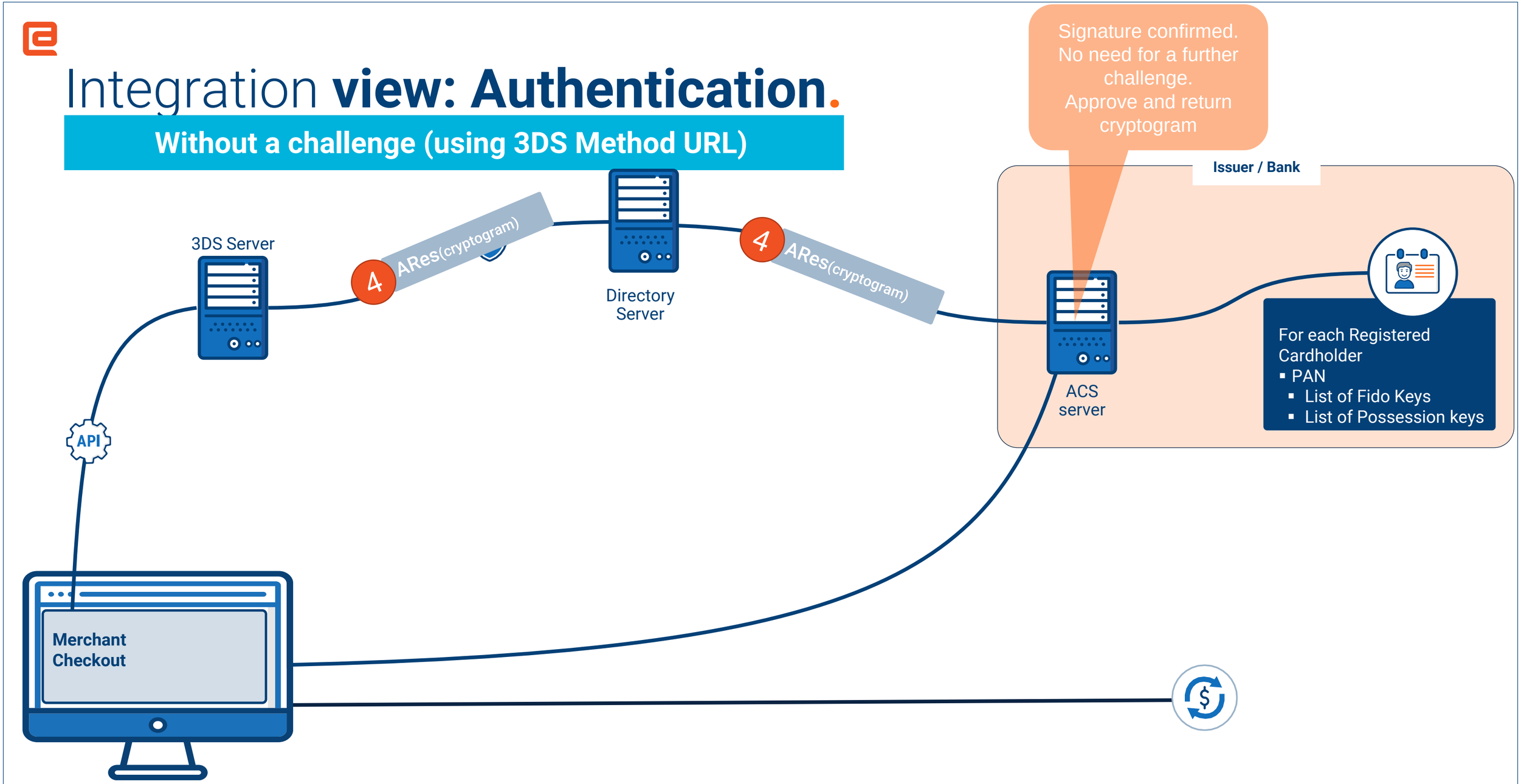
Integration **view: Authentication.**

Without a challenge (using 3DS Method URL)



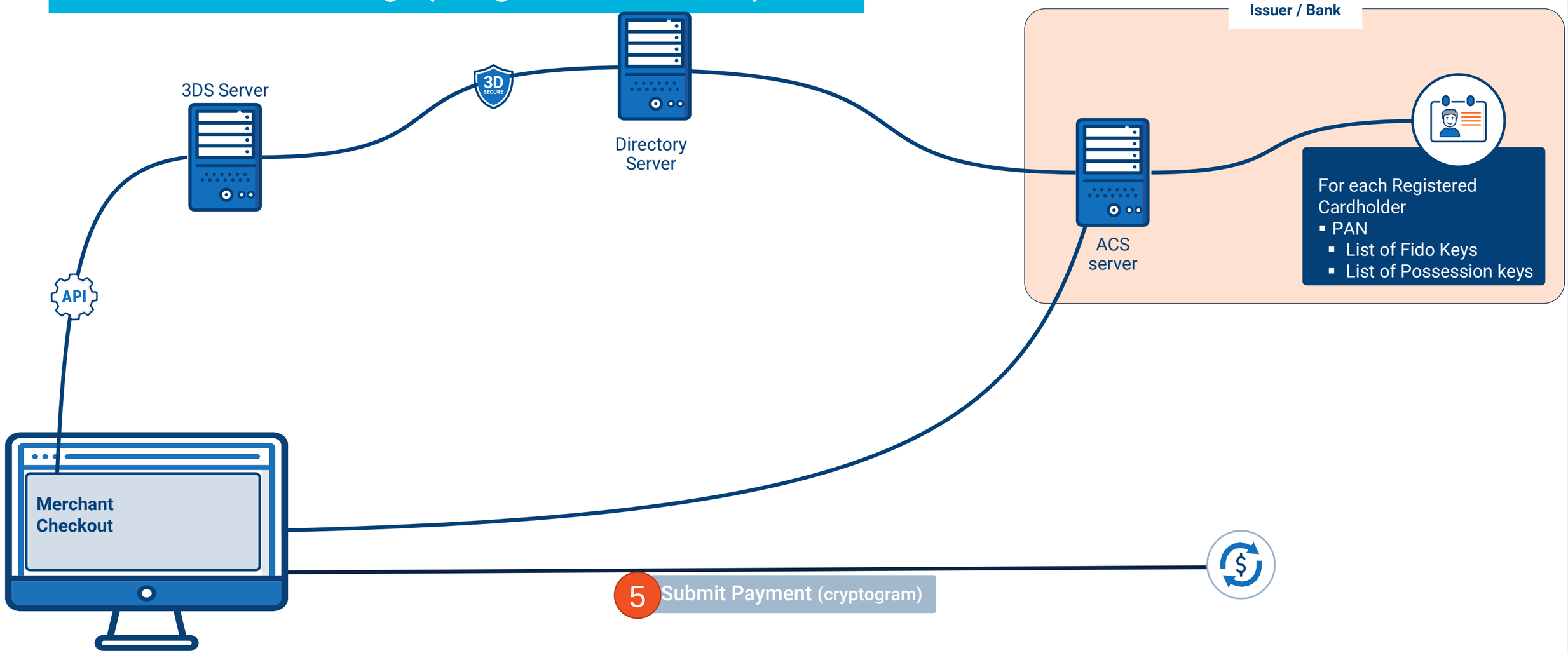
Integration **view: Authentication.**

Without a challenge (using 3DS Method URL)



Integration **view: Authentication.**

Without a challenge (using 3DS Method URL)



Impact on 3D Secure flow

- The technique aligns fully with standard 3DSMethodURL flow
  - It can work with 3D Secure 2.1 and later
- No merchant integration/modification required
  - iFrame permissions would need to support this
- If a possession credential is not available, then merchants could still revert to the SPC flow to improve challenge experience

- *Seems to align most closely to EMVCo's Browser ID requirements*
- *Might not be right to call it SPC anymore… since no sheet.*

Thank you

Discussion

Q&A

Explainer at

https://github.com/entersekt/possession-credential

Please provide comments & input

Acknowledgement

A big thank you to all those that provided input to this proposal:

- Adrian Hope-Bailie (Coil)
- Chris Dee (FIS Worldpay)
- Danyao Wang (Google)
- Ian Jacobs (w3c)
- Rouslan Solomakhin (Google)

Entersekt

The power of **trust**.