JavaScript による End-to-End セキュリティ

入門編

栗原 淳

July 29, 2019

はじめに

はじめに

この講義では

- End-to-End (E2E) セキュリティの原則
- Web サイトでの E2E セキュリティ実践のため、JavaScript での実装方法
  - ブラウザ側
  - サーバ側 (Node.js)

のさわりを学ぶ。

モダン Web サイトと End-to-End セキュリティ

Web サイトにおける昨今の情勢

- EU における General Data Protection Regulation (GDPR) の施行 (2018 年)
- GDPR に続いて、カリフォルニア、南米、オセアニアで類似の法律の制定の動き
- 日本においても、2020 年に個人情報保護法の改正法案提出の見通し

⇓

企業にとって、「正しく」「強固」に
ユーザデータ、ユーザプライバシを保護することは必須の事項

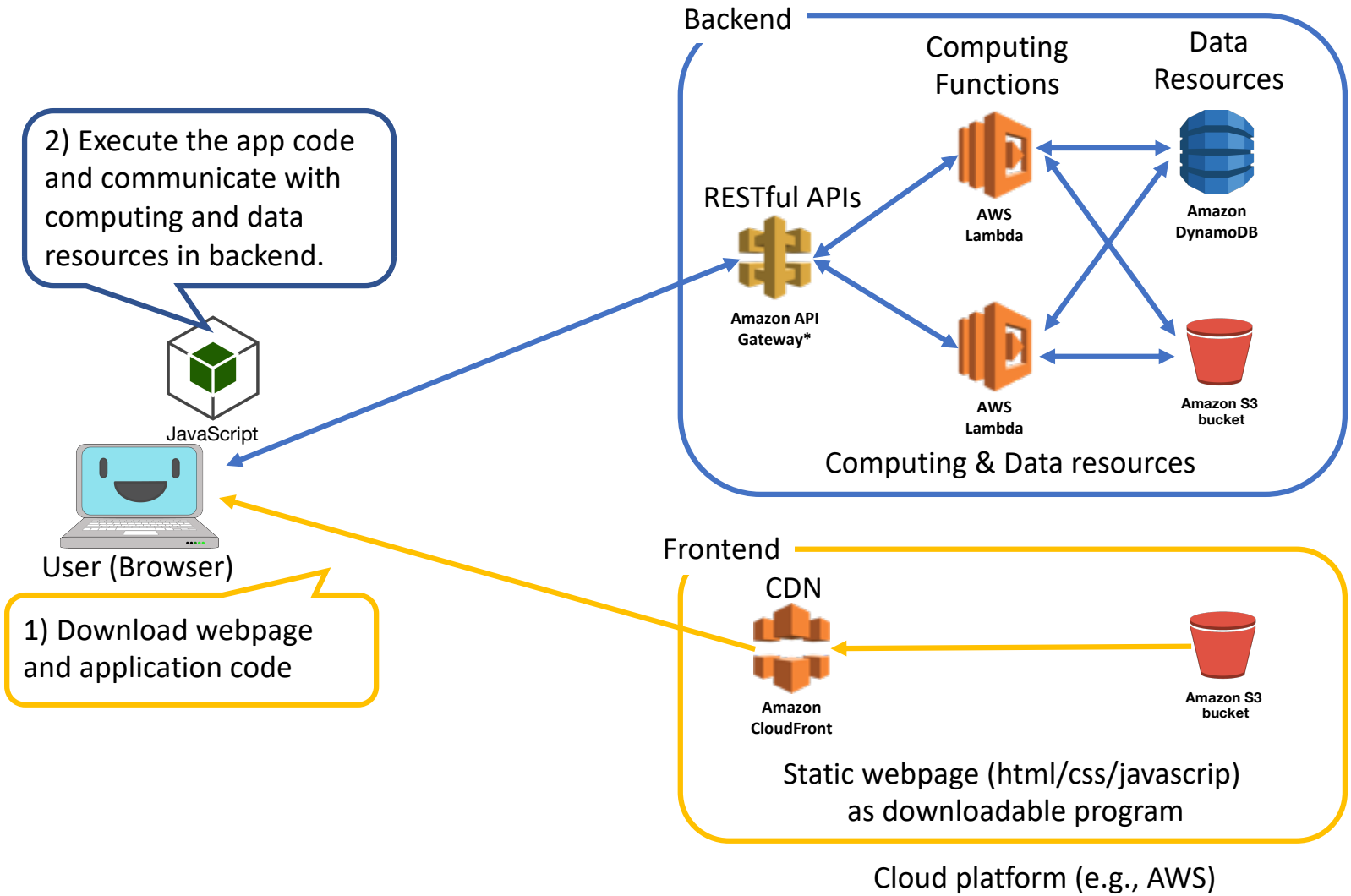
最近流行りの **Web** システム

- クラウドプラットフォーム上で構築
- 「サーバ」のない (サーバレス) 構成
- JavaScript (ReactJS など) を多用した、Single Page Application 構成

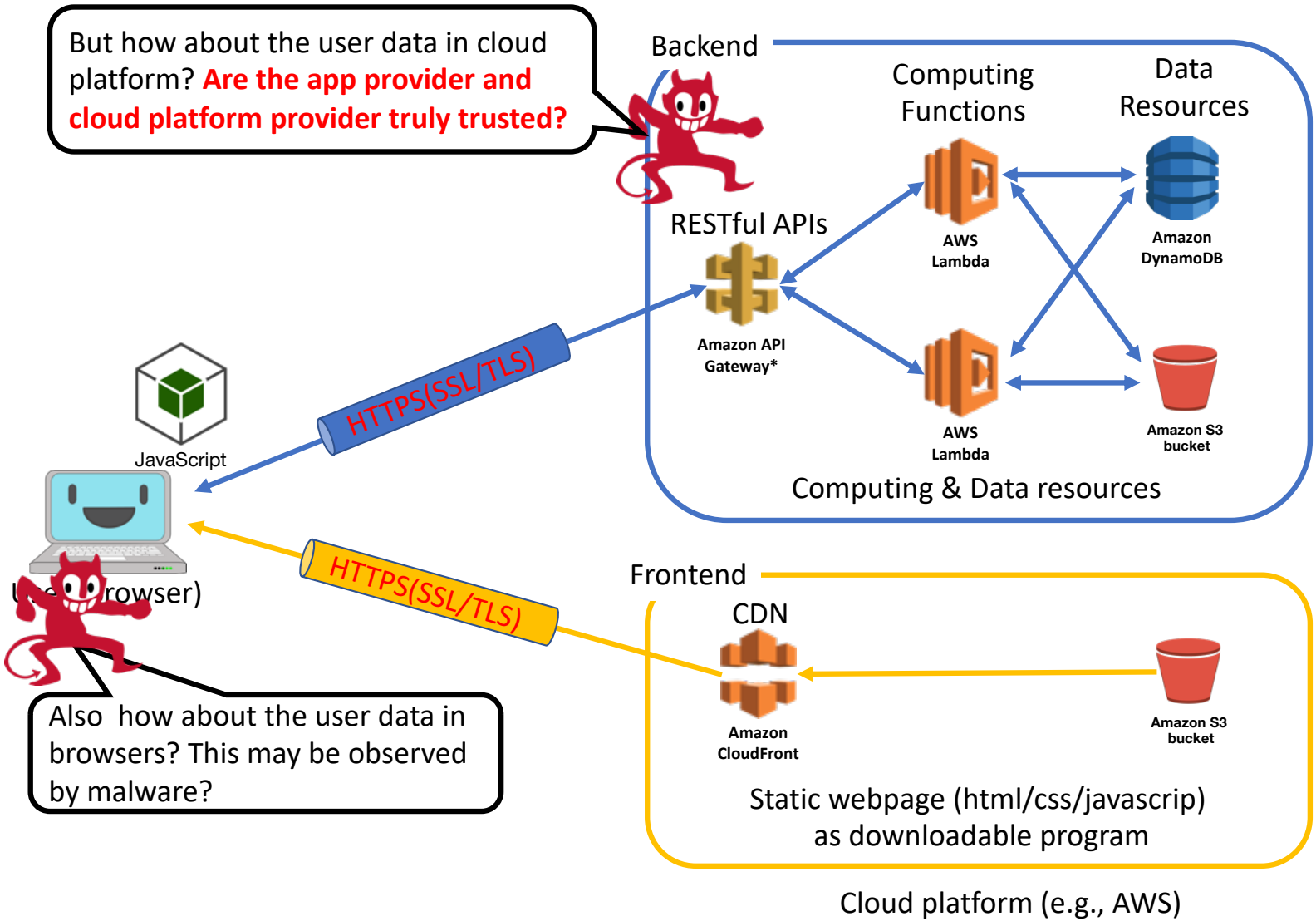
⇓

ユーザの手元で計算を実行する機会の増加

AWS を例にした典型的な構成:



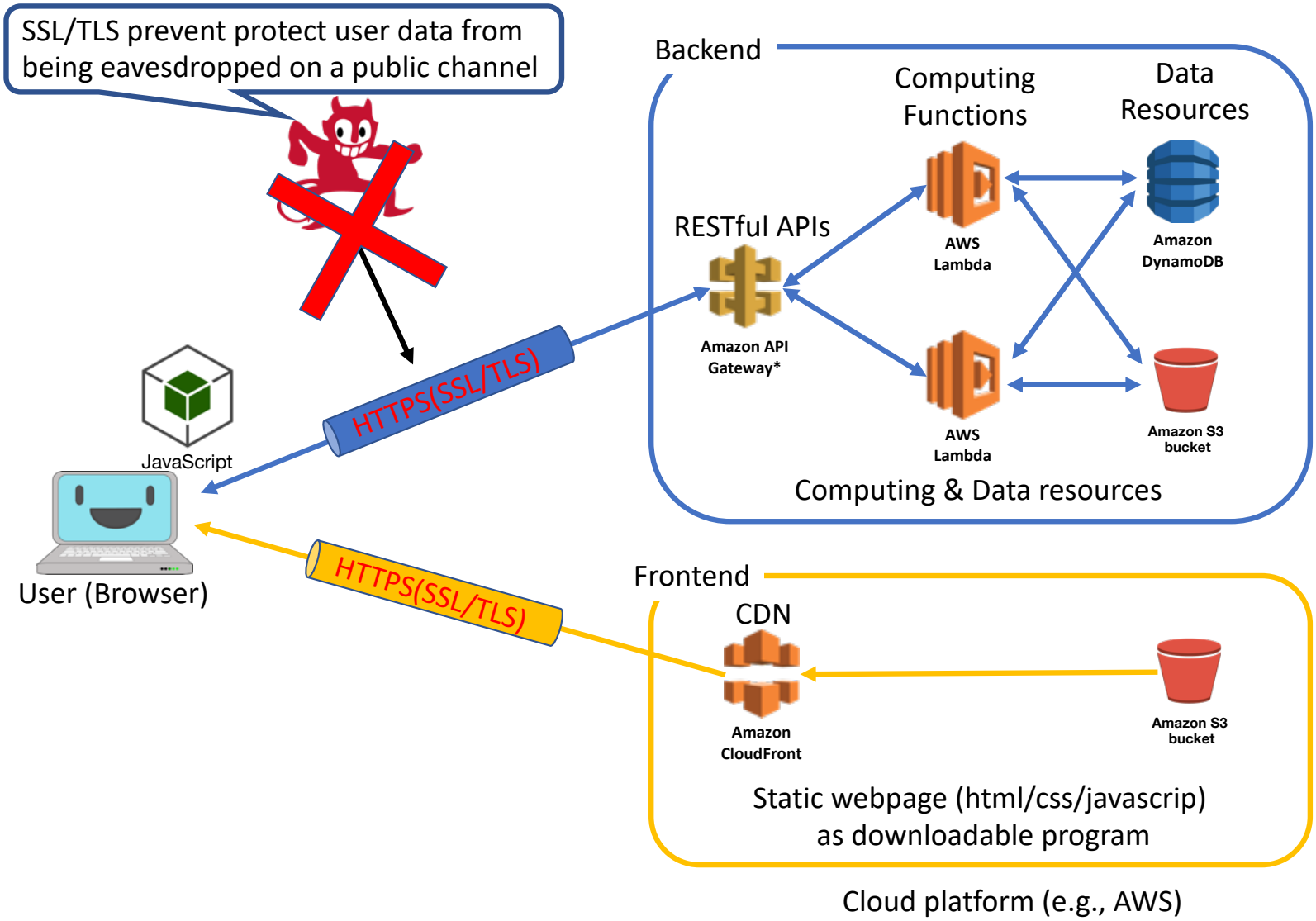
AWS を例にした典型的な構成:



通常、ユーザ・クラウド間の HTTP 通信路は SSL/TLS で保護

⇒ HTTP 通信路から外部の盗聴者へのユーザデータ漏洩を防止

AWS を例にした典型的な構成:



しかし、クラウド PF 内・ブラウザ内のデータ保護は……？

「SSL/TLS で暗号化しているから安全です」の嘘

「ストレージ暗号化しています！」
⇒ クラウド事業者に丸見えじゃないか…
⇒ 鍵が漏洩したら一網打尽

End-to-End セキュリティの原則とは

Web システムにおける End-to-End セキュリティ

導入する意味

JavaScript で暗号を使ってみよう [基礎編] 今回は AES を使ってみます。

AES とは。

ブラウザでの暗号化: **WebCrypto API**

サーバでの暗号化: **Node.js Crypto**

ブラウザ・サーバ間での相互接続性の確認

しかしサーバで復号しているのであんまり意味がない。

補足: API が違うのがめんどくさい…

手前味噌だが、統合 **API** を使って楽をすると良い

ブラウザ同士での相互接続性の確認

API を通じて暗号化データをやり取りしてみる。
E2E Security!

1. 今回は共通鍵暗号
2. 公開鍵暗号& Hybrid Encryption
3. ハッシュ・署名と HMAC
4. 超マニアック講座：RFC とアルゴリズム・フォーマット

引用文献

Appendix

This page is not counted.