Securing Your API

Shawn Wildermuth

@shawnwildermuth | wilderminds.com

This Module

Securing Your API

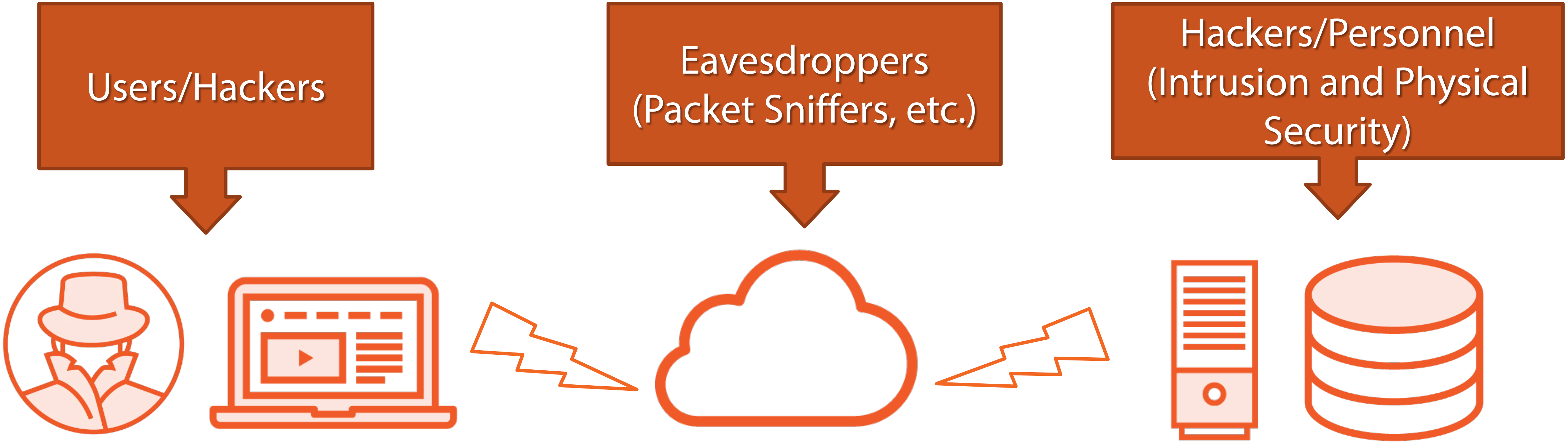
40

APIs and Security

- Do you need to secure your API?

Are you…	Secure?
…using private or personalized data?	Yes.
…sending sensitive data across the 'wire'?	Yes.
…using credentials of any kind?	Yes.
…trying to protect against overuse of your servers?	Yes.

Threats to Your API



Users/Hackers

Eavesdroppers (Packet Sniffers, etc.)

Hackers/Personnel (Intrusion and Physical Security)

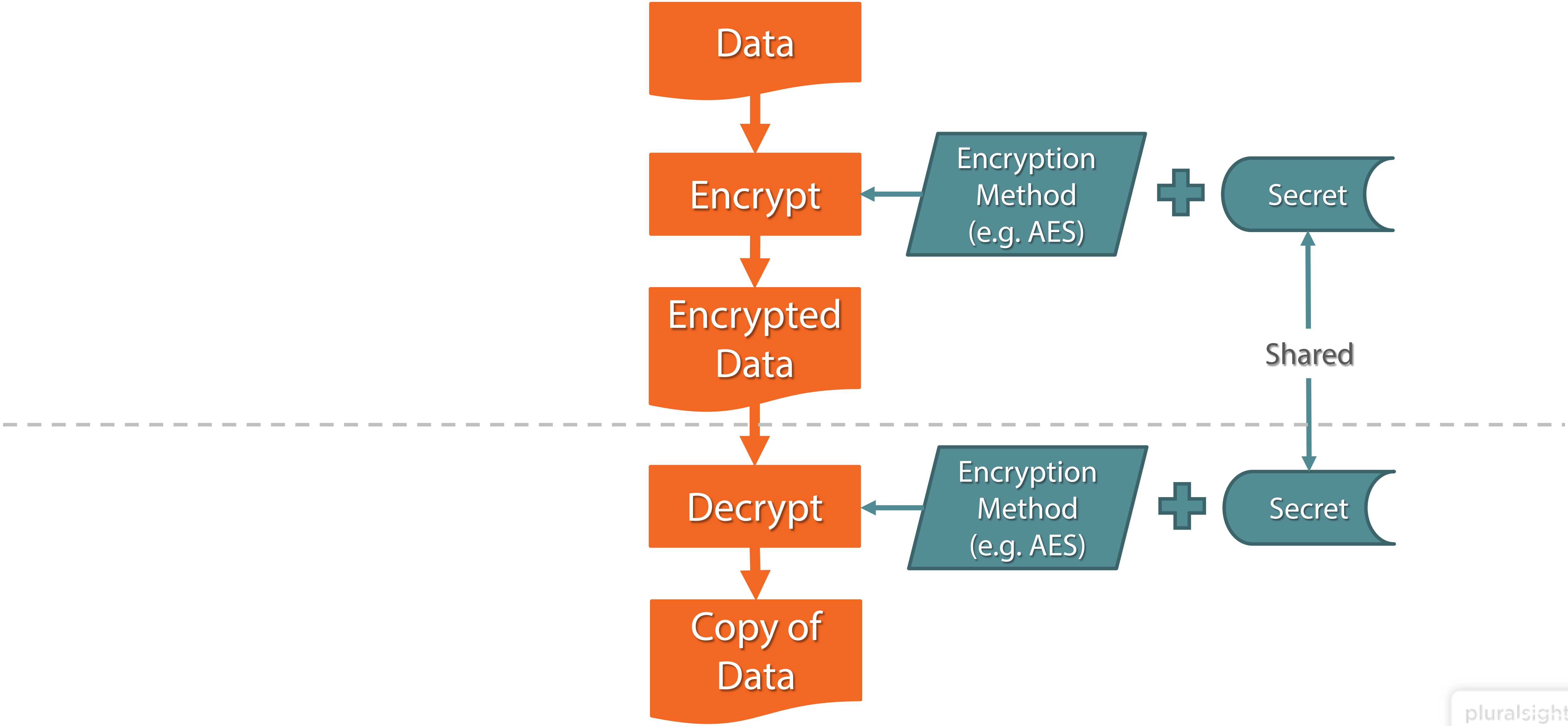
Security

- Protect Your API
  - Secure Your Server Infrastructure is outside scope of API security
  - Secure In-Transit
    - SSL is almost always appropriate
    - Cost of SSL is worth the expense…usually
  - Secure the API itself
    - Cross Origin Security
    - Authorization/Authentication

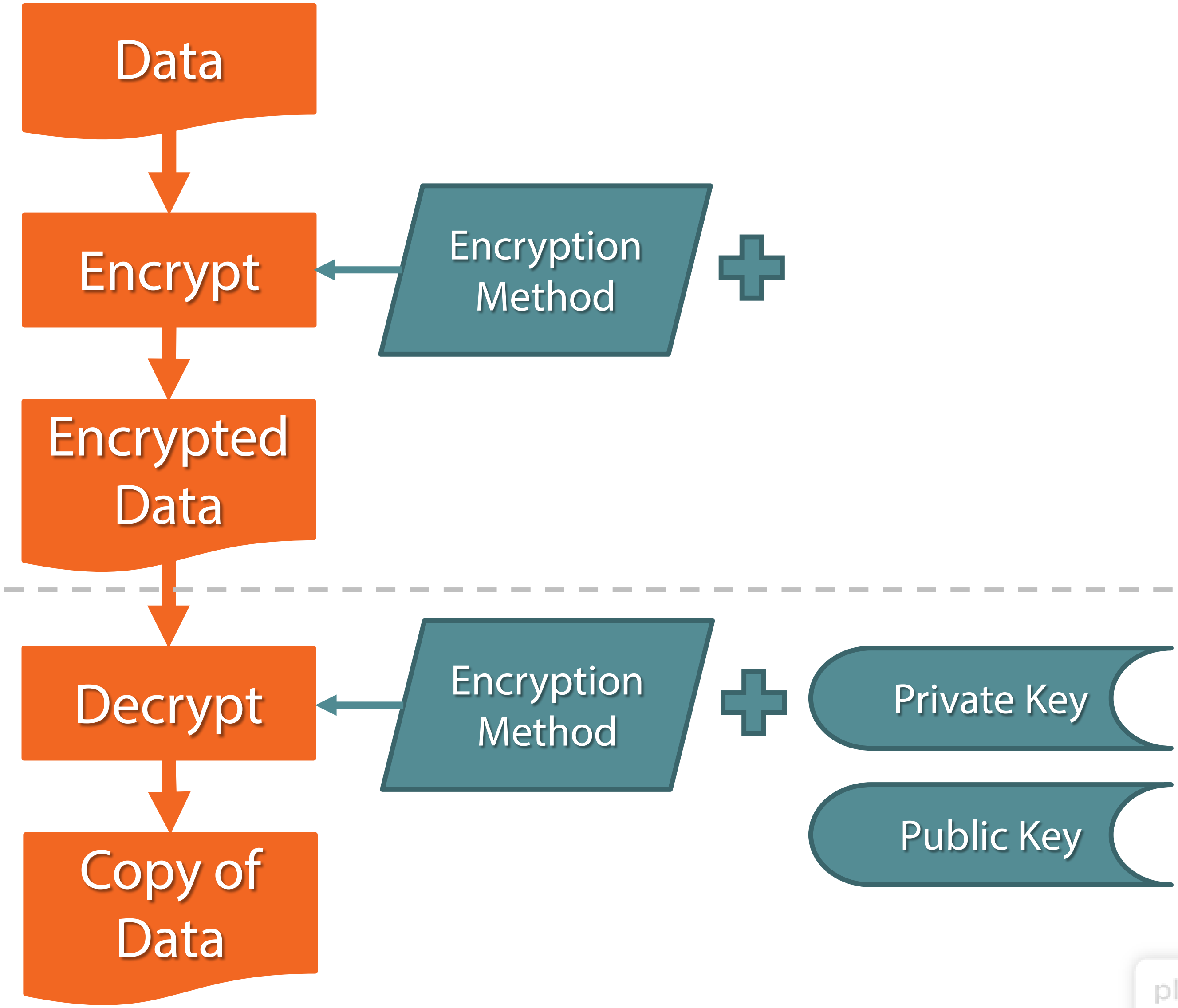
Basics of SSL

- Secure Sockets Layer
  - Used to protect communication between client and server
  - Based on Concepts of Trust + Encryption

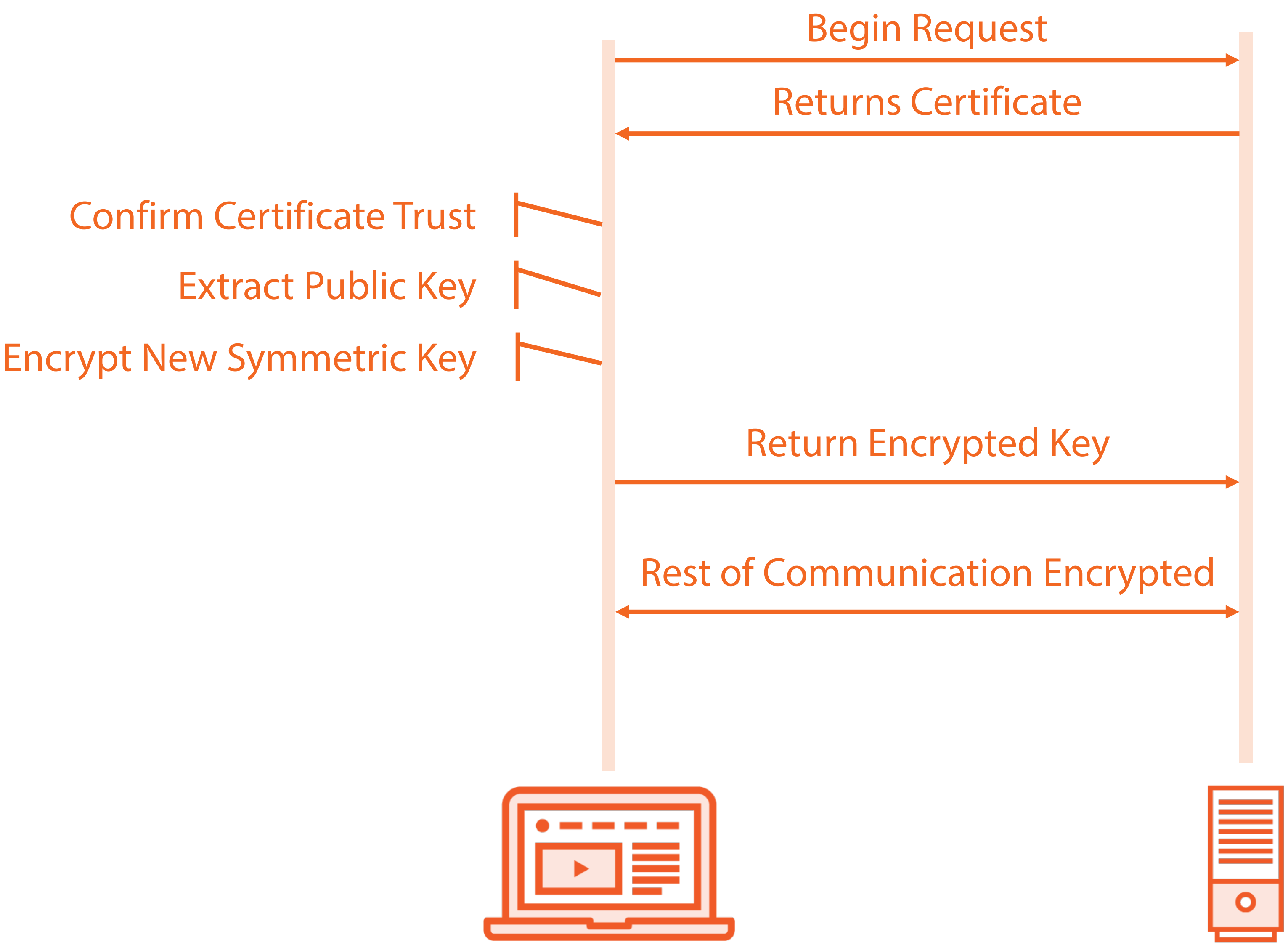
Symmetric Encryption



Asymmetric Encryption



SSL Handshake



Begin Request

Returns Certificate

Confirm Certificate Trust

Extract Public Key

Encrypt New Symmetric Key

Return Encrypted Key

Rest of Communication Encrypted

Demo

Supporting SSL

Understanding CORS

- Cross Origin Resource Sharing
  - Browsers prevent API requests across domains
  - Enable CORS gets around this limitation
    - Doesn’t affect non-browser development

Demo

Supporting 3rd Party Callers

Authentication vs. Authorization

- Authentication
  - Using Credentials to determine Identity
- Authorization
  - Verifying an Identity has rights to a specific resource

Authentication Types for APIs

- App Authentication
  - Using a secret to identify an app for your API
  - Not authenticating as the user, but as the developer!
- User Authentication
  - Accessing your API as a User

Authentication Types for APIs

- App Authentication
  - App Key + Secret is a typical scenario
- User Authentication (in order of security)
  - Cookie Authentication
  - Basic Authentication (Insecure and slow, use Tokens instead)
  - Token Authentication
  - OAuth

ASP.NET Identity

- Simple system for storing User identities, roles and claims
  - Not appropriate for App Authentication
  - Easy to do Cookie-Based Authentication
  - Basis for Basic and Token Authentication
  - For OAuth use more robust system like Identity Server*
    - Open Source
    - Great Community and Microsoft Support

`* https://identityserver.com`

Demo

Using Identity

Demo

Authenticating with Cookies

Demo

Using Identity Information

What We've Learned

Securing Your API

Security is hard…be careful implementing it

Supporting SSL and CORS is easy in ASP.NET Core

Using Identity solves simple security