APIs and NIEM

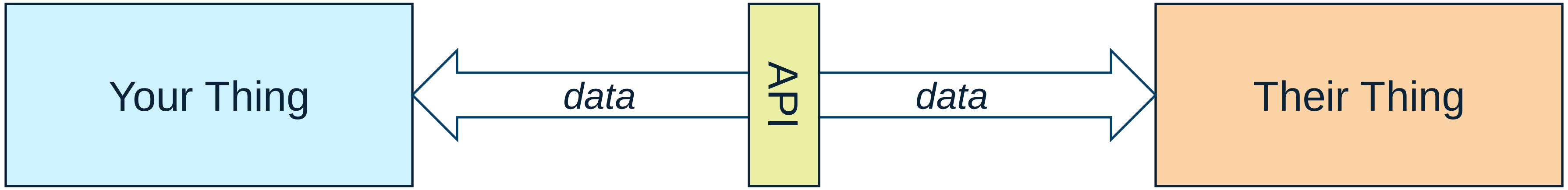
Dr. Scott Renner

19 March 2021

MITRE | SOLVING PROBLEMS FOR A SAFER WORLD™

DRAFT

Application Programming Interface (API)



- **Interface**
  - Tells the people how to interact with the thing on the other side
- **Programming**
  - Those people are software developers
  - Tells developers what they must know to write code that interacts with the other thing
  - An API is a contract between developers
- **Application**
  - Once upon a time, only end-user applications had “APIs”, by definition
  - That distinction is no longer maintained

Kinds Of API

- **Library API**
  - Tells you how to use a software library (“their thing”)
  - A form of software reuse – now you don’t have to write the library functionality yourself
  - Library code becomes part of your application

Kinds Of API

- **Library API**
- **Computing Platform API**
  - Tells you how to write code interacting with the application server, operating system, etc.
  - Your program interacting with “their thing” on the same machine

Kinds Of API

- **Library API**
- **Computing Platform API**
- **Remote API**
  - When “their thing” isn’t executing on the same machine
  - Tells you how to write code to interact with a remote resource over the network
  - Various forms of plumbing (CORBA, Java RMI, message passing, pub/sub, etc.)

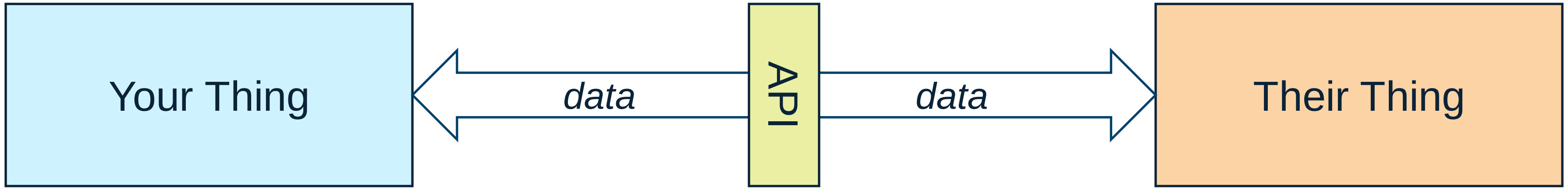
Kinds Of API

- **Library API**
- **Computing Platform API**
- **Remote API**
- **Web API**
  - A kind of remote API
  - Plumbing is HTTP / HTTPS
  - Content exchanged is usually in XML or JSON format
  - Examples: OData, OpenAPI, SOAP+WSDL

Aspects Of An API

- **Plumbing**
  - Connection between your programming language code and “their thing”
  - Parts may be fully specified, tool-generated, or written by hand
  - NIEM does not do plumbing
- **Content**
  - What data must be present, what data may be present, and what it all means
  - Documentation sufficient for the intended purpose
  - NIEM message specifications define the content part of an API
    - Directly, for APIs that exchange XML or JSON documents
    - Indirectly (at this time), for APIs with an interface definition language (Google Protocol Buffers, etc.)
- **Process**
  - An API may specify a sequence of interactions
  - Shopping cart example: Authenticate, find item, add item to cart, checkout
  - At present, NIEM does not say how to specify an interaction sequence

Abstraction and Information Hiding



- **The API presents a useful abstraction of “their thing” implementation**
  - For example: The OS and hardware present a “file system” abstraction
  - In reality, there are no “folders”, just sectors on a spinning magnetic disk
  - But you do not have to know or care about any of that
- **In theory, you do not care about the actual implementation**
- **In practice, all abstractions leak**
  - Sometimes the implementation details do matter
  - Example: file shredding tools do not work on a solid-state hard drive

“In theory, there’s no difference between theory and practice. In practice, there is”

Points Of Confusion

- **API in general / specific kind of API**
  - The concept is widely applicable, however…
  - Sometimes people say “API” meaning a particular *kind* of API