14 Septembre 2021

inflectra

13ème édition

JOURNÉE FRANÇAISE
DES TESTS LOGICIELS

API Testing, Often Overlooked,
Critical to Your Success

LE **TEST LOGICIEL**, MAÎTRISER L'ÉTAT
DE L'ART DU TEST **AVEC LE CFTL** !

ISTQB
International Software
Testing Qualifications Board

About Me

- Adam Sandman was a programmer from the age of 10 and has been working in the IT industry for the past 20 years.

- Currently Adam is a Director of Technology at Inflectra Corporation, where he is interested in technology, business and innovation.

- Adam lives in Washington, DC, USA

Takeaways

1. Why API testing should not be taken for granted, and what you can do about it

2. Overall strategies and best practices for a sound API test plan.

3. What types of API you are likely to encounter and tips for testing each of them

4. Things you can tell your developers to do, to avoid an API living hell

The API Economy

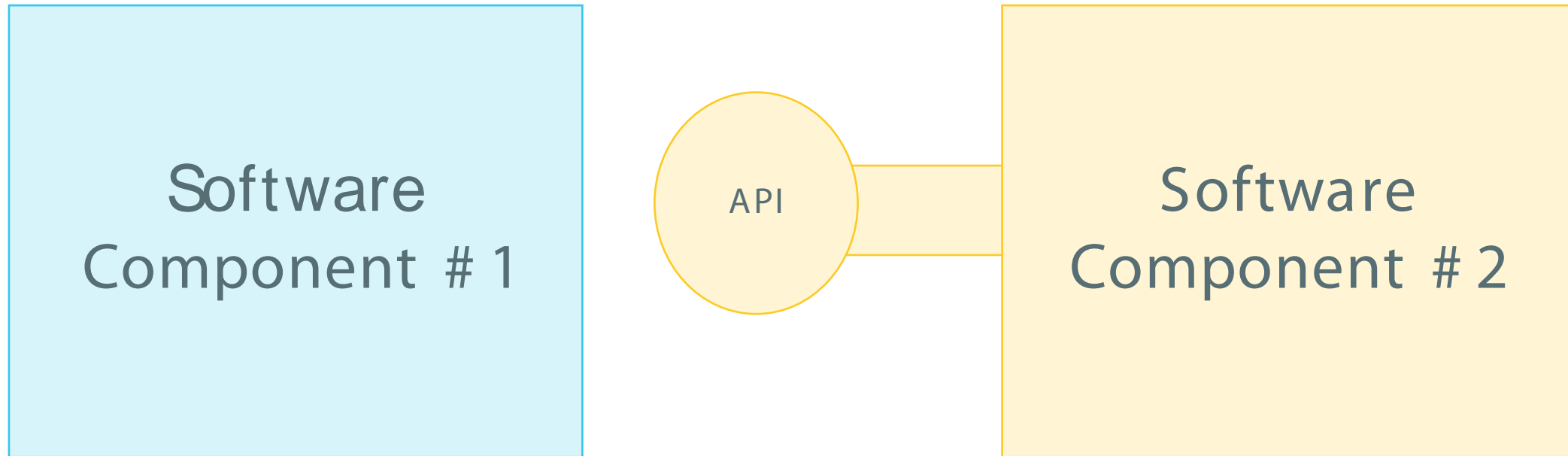Why APIs Are Vital for Today's Businesses?

First, What is an API?

API = Application Programming Interface

OR

A way for one piece of software to use the functionality of another piece of software

How Does that Work?

Software Component # 1

API

Software Component # 2

What is So Special?

- The API provides what's called an "interface"
- It acts like a contract between the software using the API, and the software providing the API
- Different people can develop their own implementations of the same API.
- Developers of Component #1 don't have to understand the inner workings of Component #2, just the implementation

Why is this a big deal?
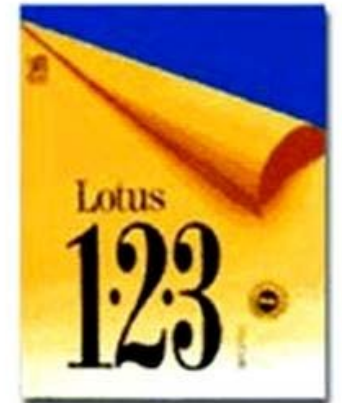
Example #1

Microsoft®

- IBM Created the "PC"
- Standard Hardware and Software
  - PC-AT, XT Hardware
  - MS-DOS / PC DOS
- The Operating System and System Specs were the API, used by Application Software
- Any hardware manufacturer could implement the same API.
  IBM no longer even makes PCs!

IBM

Example #2

- Microsoft Created the Windows API
- Any Application Developer could use this API
- However Microsoft Was able to know its inner workings before other companies.
- Do you remember Word Perfect?
- Do you remember Lotus 123
- Thanks to the Windows API No one really does!

Example #3

- Amazon Web Services (AWS)

- Anything you can do through their management interface, you can do through an API.

- Amazon went from having a cloud infrastructure built to support their "main" business, to being the dominant cloud platform in the world, thanks to APIs!

- Software companies like Microsoft, Oracle, IBM, having to play catch up...

Why Test APIs*

- MS About to Release Windows® 95
- Will Change the Computing landscape
- Testing that all applications that ran on MS-DOS will still work on Windows.
- Major Developer (SimCity) – game does not run on Windows 95 due to use of API in an unexpected way
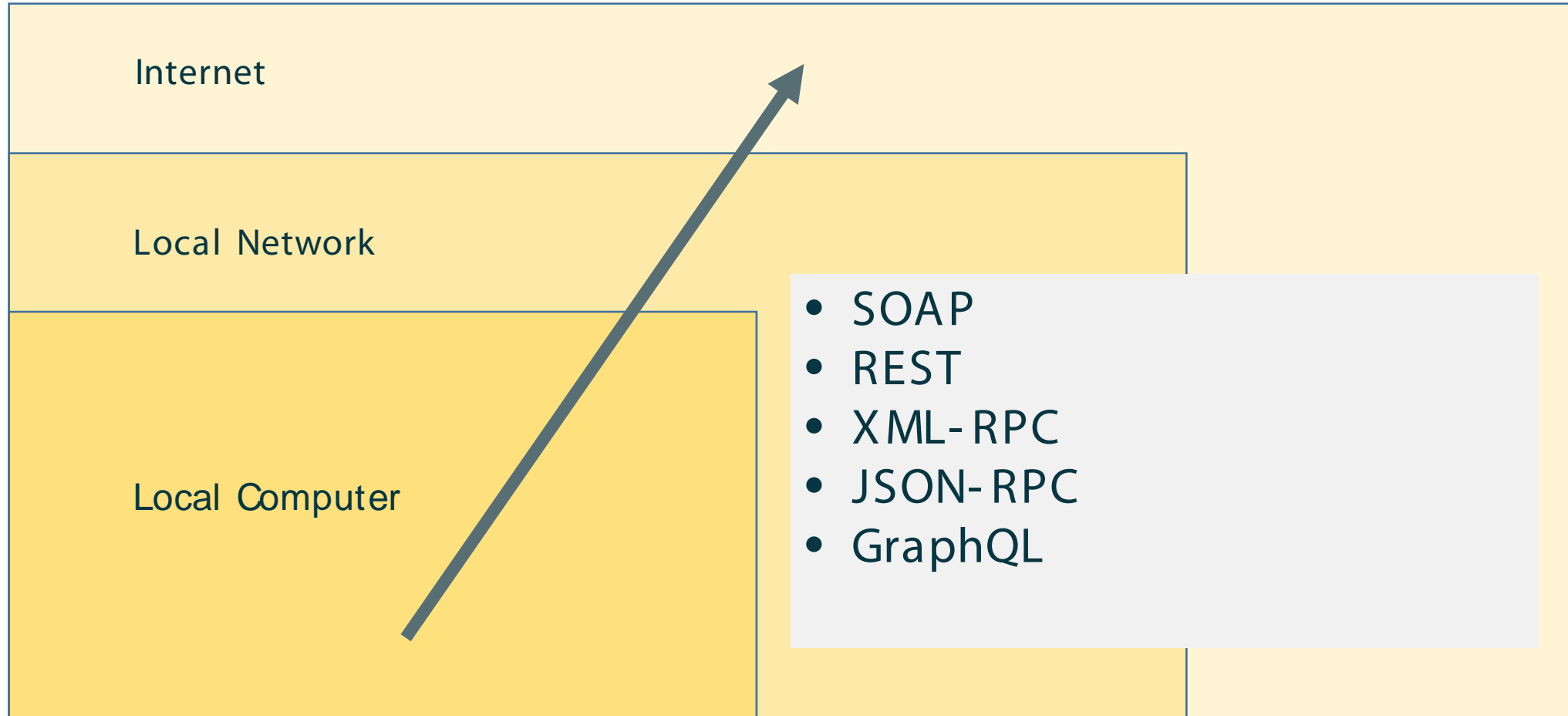- Microsoft changed the API to detect SimCity and change its behavior

*Credit to Joel Spolsky: https://www.joelonsoftware.com/2004/06/13/how-microsoft-lost-the-api-war/
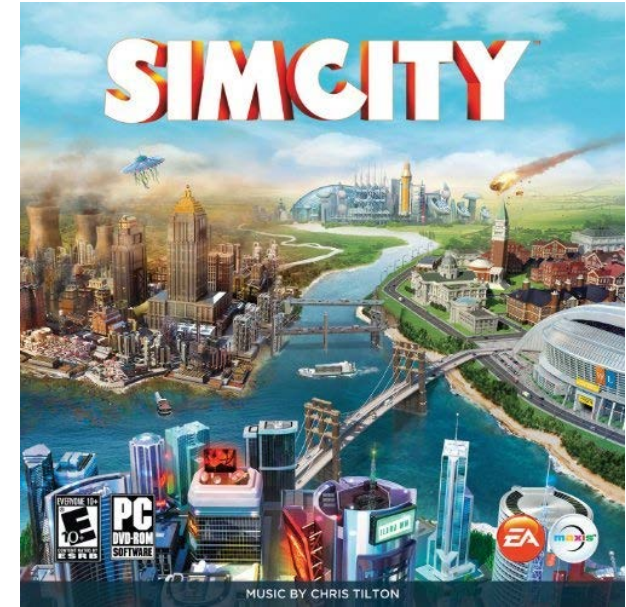
What Would You Have Done?

Types of API

Including the Past, Present and Future

APIs Become "Web Services"

Internet

Local Network

Local Computer

- SOAP
- REST
- XML-RPC
- JSON-RPC
- GraphQL

Two Families of Web Service

RPC Based

- Simple Object Access Protocol (SOAP)
- XML-RPC
- JSON-RPC

- Like calling functions.
- Self-describing
- Better for invoking actions
- Requires Tooling to use

Query Based

- REpresentational State Transfer (REST)
- GraphQL

- Better for querying and retrieving data
- Better at modelling your domain (CRUD actions)

Pitfalls of API Testing

Versions, Data Formats, Authentication, Live Systems….

Once APIs Are Released They Have a Life of Their Own

Research Necessary

- Don't assume what is being used…
- Check with customers and developers
  - Maybe that "internal API" is being used
- Sometimes undocumented behavior becomes normalized
- Atlassian
  - The GetIssue REST API would return a string or JSON, the string version was not meant to happen
  - So many developers relied on it they added a deprecation flag to the REST call to return either the old and "new" version!
  - (https://developer.atlassian.com/cloud/jira/platform/deprecation-notice-tostring-representation-of-sprints-in-get-issue-response/)

Changing the Test to Get the API to Pass = Failure in the API

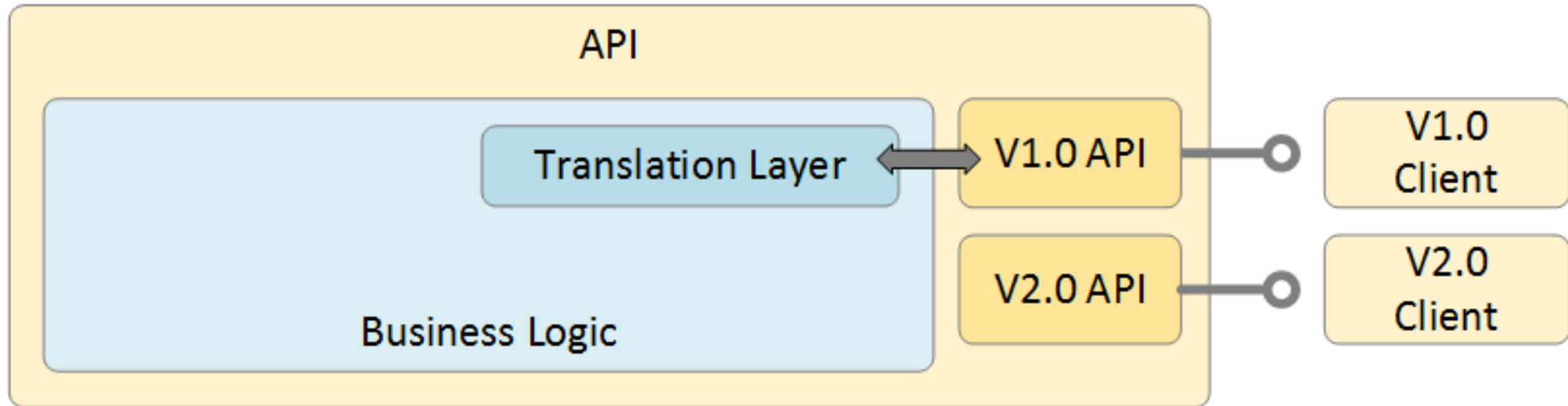(Unlike UI Testing where the UI and Tests Constantly Change)

API Versioning

- Once you have released an API into production it should never be changed
- Breaking changes should be in the next version
- You may need to write a compatibility layer?

- http://myapplication/api/v1/projects
- http://myapplication/api/v2/projects
- http://myapplication/api/v3/projects

How Do You Deal With Breaking Changes?

- Consider a translation layer:

API Versioning

- Not only the endpoints need to be versioned
  - Make sure you separate data structures into versions:

Author (v1)
First Name
Last Name
Age
Gender

Author (v2)
First Name
Last Name
Age
Gender
Race

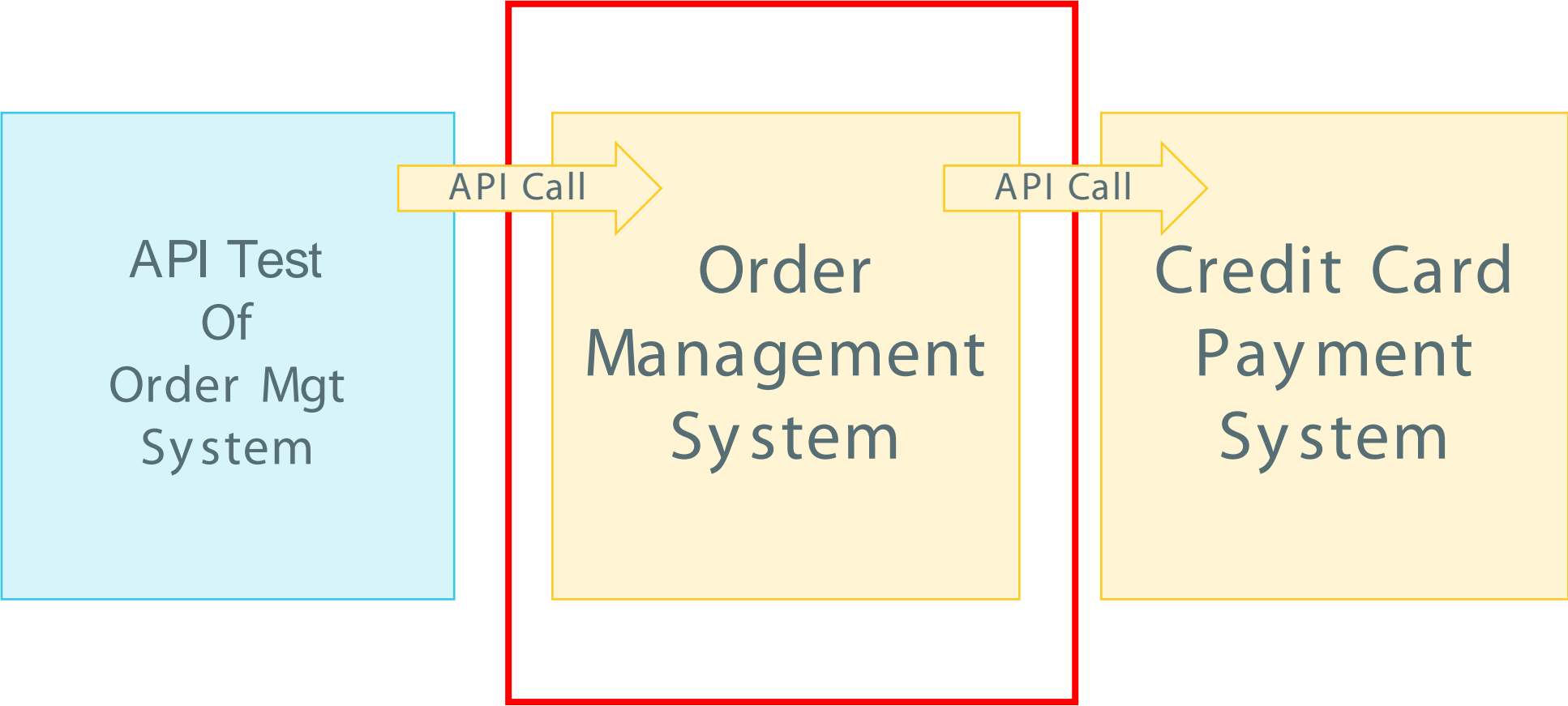| Author (v3) |
| --- |
| Full Name |
| Age |
| Gender |
| Race |
| DOB |

API Data Formats

- Some things you may need to test for:
  - Date Formats: 2012-08-01 vs. /Date(2000039843-0500)/
  - Nullable Fields vs. Empty Fields ("" vs. Null vs 0)
  - Missing Fields vs. Null Values (same thing?)
  - Enumerations (Red, Green, Blue) vs literals ("Red", "Green", etc.)
  - Binary Data (Base 64 encoded vs. Byte Arrays)
  - Unicode Characters (ëéĆĂØڤڤڤڤ)
  - Right-to-left languages (e.g. Hebrew)
  - Lengths (massively large or tiny)
  - Mismatched values (items that belong to wrong categories)

API Authentication

- Test that the API works appropriately for:
  - Authenticated Users
  - Non-Authenticated Users
  - Authenticated, but not Authorized Users
  - Authenticated and Authorized Users
  - Locked Out Users
  - Users Logged in Multiple Places
- Warning: Check that all values are appropriate for the authorization level
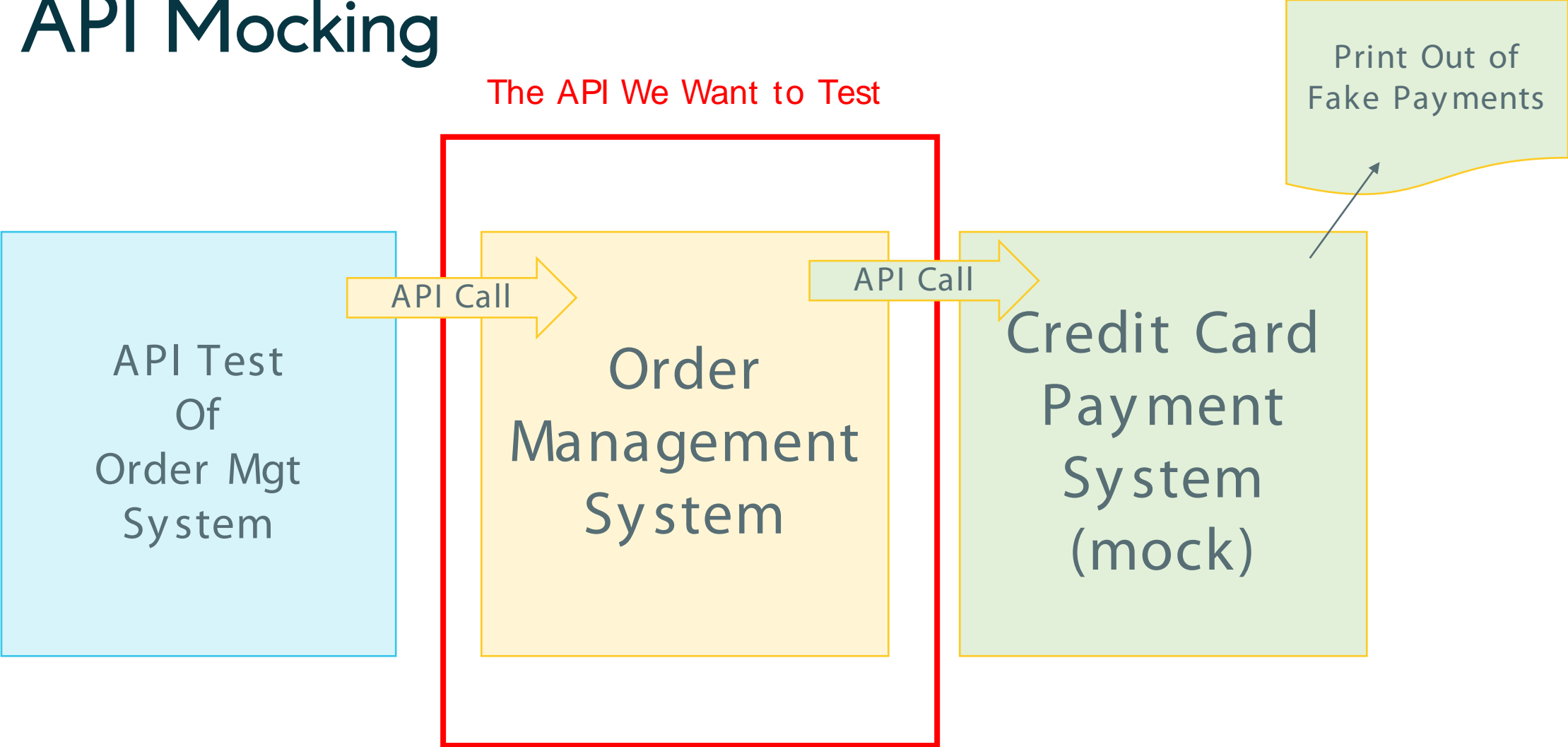 - E.g. users in project A can access/update IDs in project B ☹

API Mocking

The API We Want to Test

API Test
Of
Order Mgt
System

API Call →

Order
Management
System

API Call →

Credit Card
Payment
System

Suppose We Don't Have a Test Credit Card System, What Should We Do?

API Mocking

The API We Want to Test

API Test
Of
Order Mgt
System

API Call →

Order
Management
System

API Call →
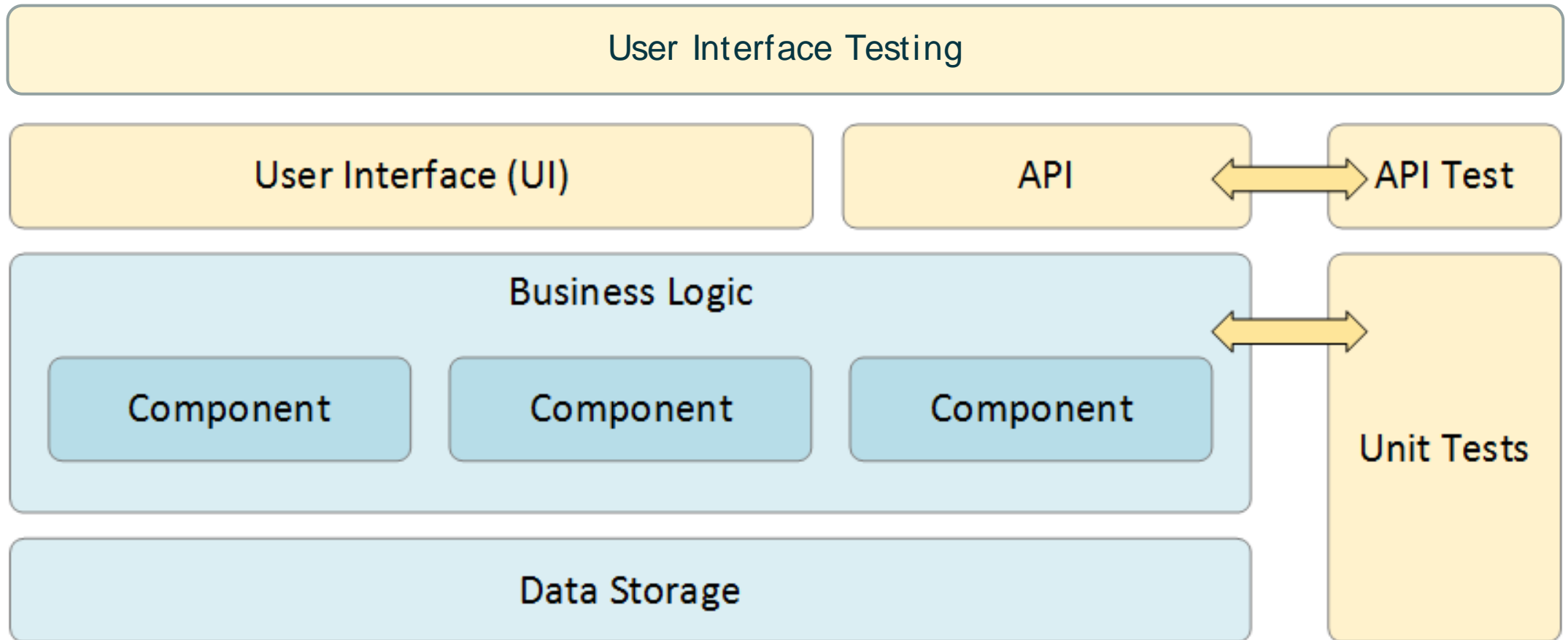
Credit Card
Payment
System
(mock)
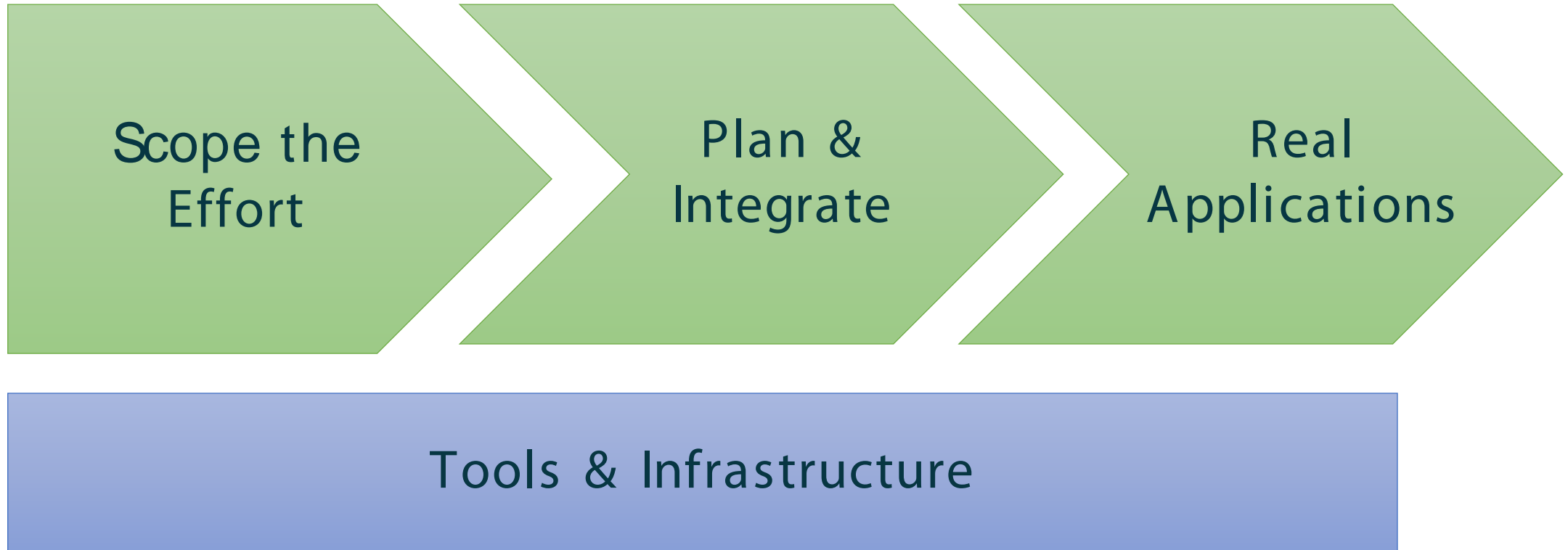
Print Out of
Fake Payments

Creating the API Test Plan

Including the Past, Present and Future

Where Does API Testing Fit In?

Elements of the Plan

Scope the Effort

- Identify the API Endpoints
    - How Many Versions (v1, v2, v3)
    - How Many Types (SOAP, REST)
    - How Many Formats (XML, JSON)
- How far back do you support?
- What API Endpoints are most widely in use?
    - Deep, robust tests on 20% most widely used
    - Broad smoke tests on remaining 80%
- Sample Applications that can be used to scope coverage

Plan and Integrate

- Developers write basic API tests for all coded endpoints
- Should cover all operations for all versions in all supported data-formats
- New APIs Should Have Unit Tests Written in Parallel
- Testers / other developers write additional, deeper tests for the most widely used APIs
  - This requires testers who are familiar with code and APIs
- Integrate into CI pipeline along with non-API tests
- Mock any dependent systems that cannot be run in CI

Real Applications

- Don't just rely on coded API tests
- Get real applications / API clients
- Try them out with all versions of the API
- Get old versions of the applications that could be in use
- You will find things that the API tests miss ☹

Tooling

- Some tools we have found useful:
  - SOAP API Tests
    - SoapUI
    - Rapise
    - Visual Studio and NUnit
  - REST API Tests
    - Postman
    - Rapise
    - Apigee
    - Rest-Assured
    - JMeter (Load Testing)

Wrap Up

- API Testing is critical for any software platform business, and very important for almost all organizations

- API Testing requires an understanding of how the API is being "used in the wild"

- You need to work with the product owners and developers to scope the API testing effort appropriately

- Testing with dedicated, automated API tests is necessary, supplemented with real testing using actual client apps

Questions?

Thank you for attending this workshop/session.

Please fill out an evaluation form.