An Introduction to:

Testing in the API Economy & Why It’s Critical to your Organization

A PUBLICATION OF APICA

TABLE OF CONTENTS

INTRODUCTION

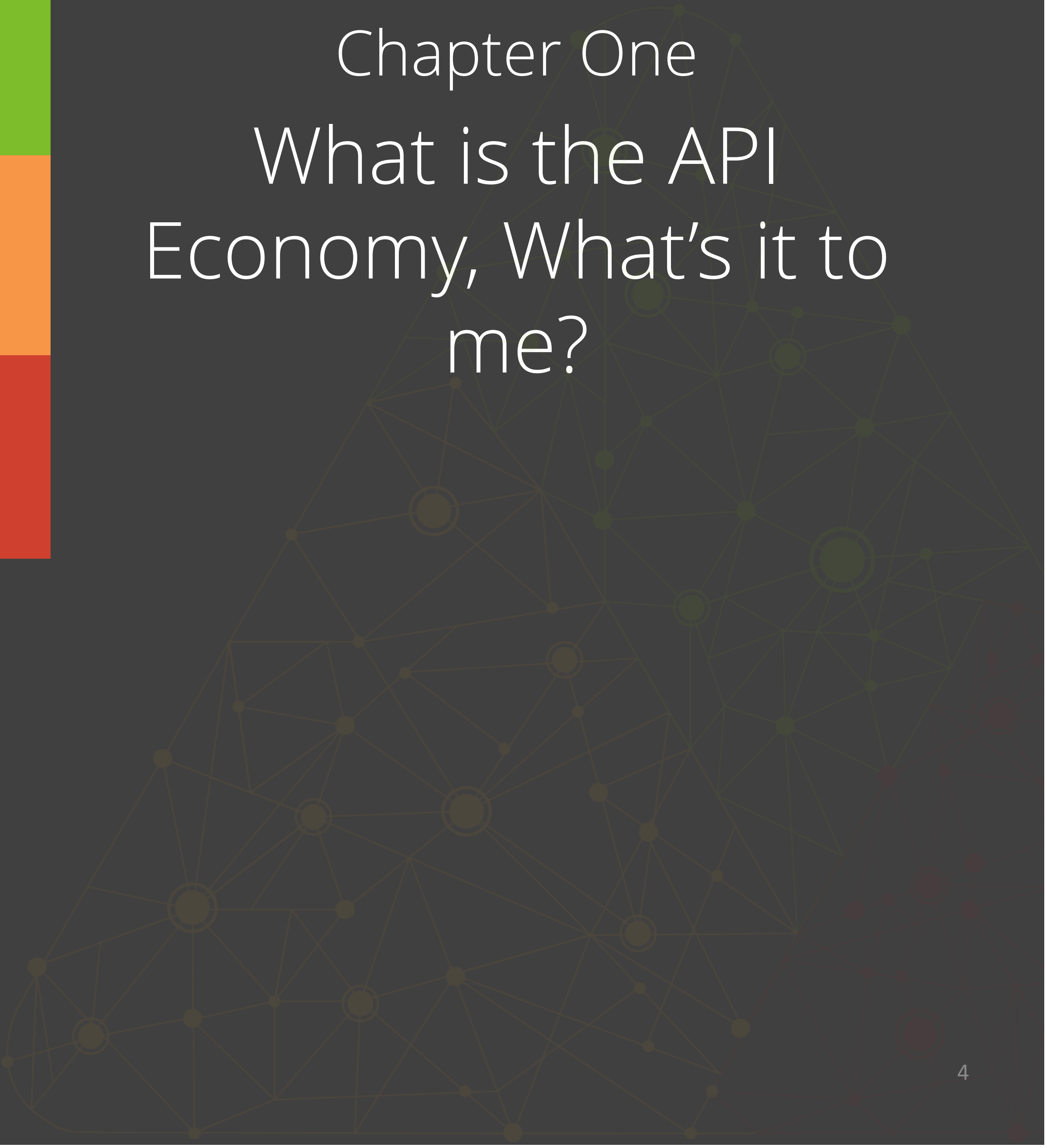
Succeeding in the API Economy

APIs used to be only for developers, now they're an enabler for turning for a business or organization into a platform. Your business's products and services can be reworked and implemented in inventive ways to generate new revenue streams. Succeeding in the API Economy is essentially the same as succeeding in the Data Economy—and many businesses find it helpful to treat the API itself as the new product. This may seem a bit unusual since the API shares a lot of assets with another product, but both need to stand alone. Google Maps and the Google Maps API, for example, are two different products that share some of the same primary services.

A well-implemented API will expand on an existing product or concept, which in turn will seize new business growth opportunities. As a general rule, an API should improve performance—making it easier, for instance, for a business to include a map on their website, or for a homeowner to change thermostat settings remotely. However, APIs require a solid, reliable Internet infrastructure. If that IoT thermostat doesn't get the message that it should turn off the air conditioning because the server that handles the communication is overloaded, the customer will be less than thrilled. Latency is also important: The communication needs to happen quickly or the customer may give up. Load testing services are essential to making sure your infrastructure is ready to handle communication for your API.

Chapter One

What is the API Economy, What's it to me?



What is the API Economy?

According to IBM, the API Economy is "the commercial exchange of business functions, capabilities, or competencies as services using web application programming interfaces." The Internet makes it possible for devices and people all over the world to be connected and APIs are the means applications use to reach outside of themselves and exchange information over those connections.

Examples of APIs

- When purchasing movie tickets online, an API is used to verify credit card information
- Booking a plane ticket on Orbit, Kayak or Flygstolen.se uses APIs to query 100's of sites.
- Controlling Alexa through your Amazon Echo
- Uber uses google maps through an API to connect drivers with a car riders

The API economy is an enabler for turning a business or organization into a platform.

Why Does it Matter?

API ECONOMY 2016 – THE BEGINNING

As organizations and the public continue to grow more comfortable with implementing API technology in day-to-day activities, the market for API products continues to expand.

If you used your Facebook credentials to log in to a Disqus comment board today, or used a Google maps widget to locate a store you're visiting later, you've already used two different APIs. APIs are also helpful beyond web browsers. For example, if you ordered more Bounty paper towels this morning with an Amazon Dash button, or turned down your home's Nest thermostat from work using a mobile app, you're already familiar with Internet of Things devices running APIs.

API ECONOMY 2017 – THE NEW BUSINESS ENGINE

Prediction: Due to the trend of organizations running as API engines and increasing API use and their adoption being used to drive many parts of business, we will aggregate a substantial number of functions into APIs - driving the internet economy. To ensure these combined API's deploy and function properly, application creators will lean more heavily on visibility and testing solutions.

Secret Sauce for Digital Platforms

“Application Programming Interfaces (APIs) and Software Development Kits (SDKs) are the secret sauce for digital platform innovation and enhancement. For example, early in its online presence Netflix opened public APIs from which users created links to New York Times movie reviews and ways to discover niche content. An API can give your best users the chance to better customize and personalize your platform to their needs. APIs and SDKs give users the permission and the power to create value for themselves.”

- Michael Schrage

Chapter Three

Testing in the API Economy

SOAP vs. REST API Testing

Simple Object Access Protocol (SOAP) and REpresentational State Transfer (REST) are protocol and web service specifications, respectively, that APIs can use to exchange information between systems. The main difference between how the two operate in the testing process is REST's ability to cache data requests (which SOAP cannot do). The tests need to be configured differently to simulate real-world use cases because REST platforms can quickly recall information from the cache, whereas SOAP platforms need to make a new data request with each interaction (resulting in more transactions and higher latency). This difference between the two standards can produce dramatically different communication traffic trends.

Your QA testers will need to utilize different configurations and tools depending on which standard is used. SOAP tests may need to pay more attention to interoperability aspects than REST tests as well.

Functional vs. Manual API Testing

The primary difference between functional and manual API testing is philosophical. Functional API testing involves determining if all the necessary software features are present and operational; manual API testing, on the other hand, involves manually testing those features for defects. Both are important parts of the API testing process.

Testing software can help streamline the process for both functional and manual tests, but each test will need to be programmed differently to meet the test criteria. The functional test will need to be programmed simply to determine whether all the software's required features work, whereas the manual tests will need to be designed to exercise features using multiple test cases, to probe for errors.

API Testing vs. Unit Testing

Unit tests are designed to determine if a part of the program works correctly. API tests for each sub-system/platform are done before testing the whole system to isolate the weak point in the system. The tests operate on two different levels of the overall software testing process: The unit test needs to be completed before running the API tests, which are a part of the integrations testing process.

The development team usually handles the unit testing work, while API testing responsibilities fall to the Quality Assurance staff. If the QA staff runs API tests before the unit testing is complete, the API test data will be deemed invalid in the event that the unit testing process finds problems with the software. The test will need to be redone after the development team fixes the identified errors.

Chapter Two

What Types of Bugs Does API Testing Find?

WHAT TYPES OF BUGS DOES API TESTING FIND?

- Secure that the API can deliver the expected throughput
- Verifies that the system resources on the API platform can handle the load
- Unused flags
- Missing or duplicate functionality
- Reliability Issues. Difficulty in connecting and getting a response from API.
- Security Issues
- Multi-threading issues
- Performance Issues. API response time is very high.
- Improper errors/warning to caller
- Incorrect handling of valid argument values
- Response Data is not structured correctly (JSON or XML)

Chapter Four

Automating API Testing

Automating API Testing

While it's possible to perform a stress test on RESTful APIs by simply gathering a group of people together using multiple devices to mass-generate HTTP requests, it's not often practical. When designing a RESTful API stress test, your development team will likely single out a specific page request that makes a REST call for the purpose of the test. Automating this process can save your business time and energy.

Your development team can utilize a browser automation program or HTTP request tool to simulate a single user functional test. A browser automation program like Selenium records user actions that can be exported and reused as a script on multiple devices simultaneously, which simulates virtual users repeatedly calling a RESTful API request. Alternatively, try Google Chrome plug-in RESTful Stress to stress test an API by batch invoking HTTP requests.

Send from Multiple Systems

Running the test on a single device may not generate enough HTTP requests to push the API as far as it needs to go. A browser action recording program will work on any platform, but is limited by how quickly the computer can simulate the test (which can be offset by things like page load times and database calls). Since the stress test is not examining these parts of the application infrastructure, those bottlenecks can hinder the actual test. You can work around these limitations by running the test on multiple computers at once.

Automating API Testing

Bringing It Together, Testing in Development

Looking at a hypothetical ASP.NET application for the test, the development team may opt to create a large testing load by using multiple computers running browser recording software. Running these tests can also be a helpful measure of how well the API performs under heavy traffic loads by utilizing the testing or development servers instead of the production servers. That's because the test is looking at whether the API is able to keep up with high volume requests.

Chapter Five

3 Common Problem-Solution Use-Cases

3 Common API Testing Problems & the Solutions

1. Problem

The growth of API usage is so rapid that systems can’t keep up. Companies are forced to add additional infrastructure and limit how many calls are made per user/company.

Solution

Always test for expected growth, rather than current environment.

2. Problem

The API calls often produce insufficient or incorrect data, which breaks integration levels, delays sites or delivers unresponsive sites. For example, have you seen your site “time-out” when you’re streaming a show? Enterprise streaming companies are seeing massive communication problems between their back-end API and the database. PokemonGO is another example. There were so many requests happening that the response time was significantly slower.

Solution

Create multiple calls, rather than just one. If you integrate synthetic monitoring with APM tools, you can find the problem early on and re-write/re-define the query by breaking it into smaller chunks.

3 Common API Testing Problems & the Solutions

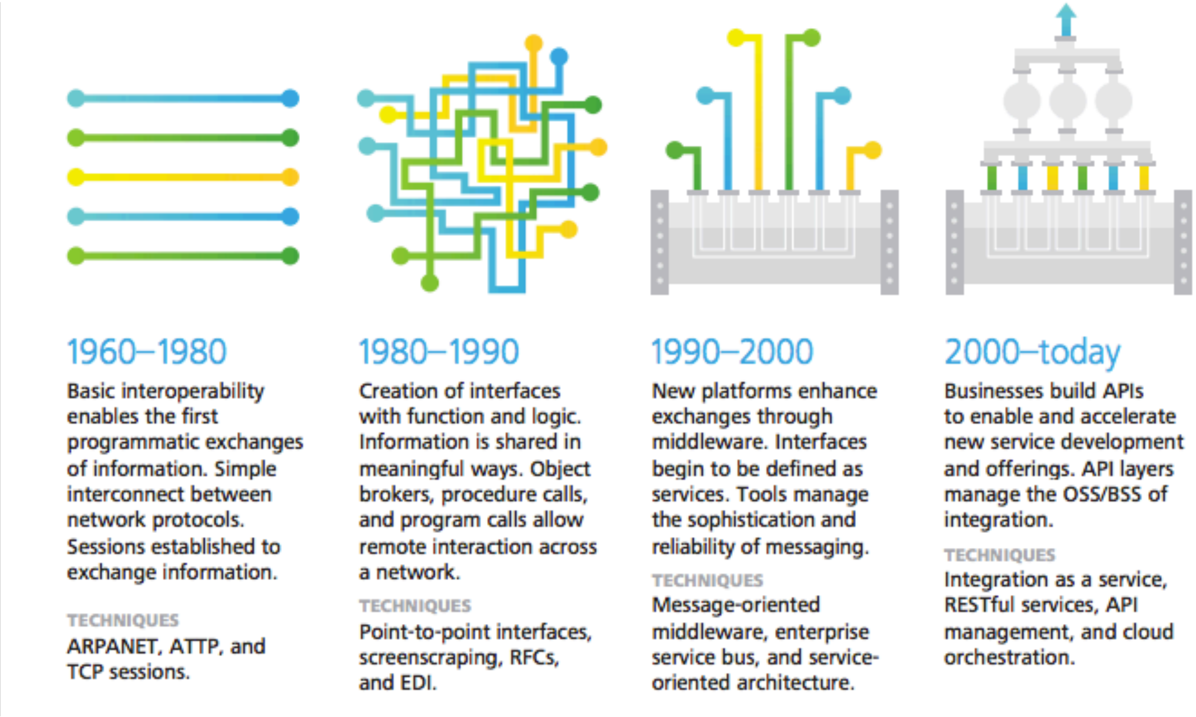
3. Problem

Consumer-based use and growth of mobile applications and IoT systems is higher than companies can keep up with. Development is rushing releases, causing security and feature errors (from an API perspective).

Solution

Aggressively test throughout development lifecycle, starting in design phase, monitoring API usage and functionalities. This can be done with a synthetic monitoring solution.

The Evolution of API’s, by Deloitte



1960–1980

Basic interoperability enables the first programmatic exchanges of information. Simple interconnect between network protocols. Sessions established to exchange information.

TECHNIQUES

ARPANET, ATTP, and TCP sessions.

1980–1990

Creation of interfaces with function and logic. Information is shared in meaningful ways. Object brokers, procedure calls, and program calls allow remote interaction across a network.

TECHNIQUES

Point-to-point interfaces, screenscraping, RFCs, and EDI.

1990–2000

New platforms enhance exchanges through middleware. Interfaces begin to be defined as services. Tools manage the sophistication and reliability of messaging.

TECHNIQUES

Message-oriented middleware, enterprise service bus, and service-oriented architecture.

2000–today

Businesses build APIs to enable and accelerate new service development and offerings. API layers manage the OSS/BSS of integration.

TECHNIQUES

Integration as a service, RESTful services, API management, and cloud orchestration.

> *“To ensure these combined API’s deploy and function properly, application creators will lean more heavily on visibility and testing solutions.”*
>
> Sven Hammar, CSO, Apica

CONCLUSION

It's critical for your organization to utilize testing in the API economy to ensure you remain a competitive business moving forward.

If you'd like more information on the API economy and testing and synthetic monitoring services for your APIs, check out our blog here or join us for an educational 30-minute webinar, API Basics for Big Data Analysis on Thursday, February 23rd, 2017 at 11AM PST.

RSVP FOR WEBINAR