

Applying Domain Driven Design to APIs and Microservices

James Higginbotham
@launchany



**Your API tells a story about your
company and your vision**

Let's get started

Our new generation of customer experiences are built with APIs, and we're opening them up to you.

What problems can we help you solve?



SwiftID

Our SwiftID API is a two-factor authentication product that allows you to give your customers an easy, secure way to approve access requests for confidential information. Better than a social identity, SwiftID validates the person on the other end based on trusted data you'd expect from a bank.

[Learn More](#)
[Documentation](#)



Rewards

Our Rewards API allows you to get information on the miles, points or cash rewards your customers have earned with their Capital One accounts. Based on their individual rewards balances and redemption opportunities, you can tailor your digital experience in real time.

[Learn More](#)
[Documentation](#)



Credit Offers

Our Credit Offers API returns a personalized list of Capital One credit card offers in under 60 seconds based on just a few pieces of personal information. You can use this API to provide a valuable service for your customers, and as an affiliate you benefit too.

[Learn More](#)
[Documentation](#)

IBM Watson Developer Community

Your community for dev support, inspiration, and the latest developer news about IBM Watson.



Learn how to get started



Ask questions, find answers



What's new in Watson

[LATEST NEWS](#)[COMICS](#)[MOVIES](#)[VIDEOS](#)[GAMES](#)[TV](#)[CHARACTERS](#)[DEVELOPER PORTAL](#)[How-Tos](#)[Interactive Documentation](#)[Get a Key](#)[Help](#)[News and Updates](#)

CREATE AWESOME STUFF WITH THE WORLD'S GREATEST COMIC API

The Marvel Comics API allows developers everywhere to access information about Marvel's vast library of comics—from what's coming up, to 70 years ago.

[GET STARTED](#)

**A great API design strategy is
critical for API product adoption
and sustainable application
development**

Feature-packed payments

A suite of APIs that powers commerce for businesses of all sizes.



Built for developers

Simple, composable,
orthogonal APIs.



Powerful toolkit

Works with any type of
business.



Perfectly scaled

Robust support for
companies of all sizes.



Seamless security

Trusted, secure, and
headache-free.



Built for developers

Robust, clean APIs that let you focus on building great products.

curl

Ruby

Python

PHP

Java

```
1 curl https://api.stripe.com/v1/charges \  
2   -u sk_test_BQokikJ0vBiI2HlWgH4oIfQ2: \  
3   -d amount=400 \  
4   -d currency=usd \  
5   -d "description=Charge for test@example.com" \  
6   -d "source[object]=card" \  
7   -d "source[number]=4242424242424242" \  
8   -d "source[exp_month]=12" \  
9
```

These actually work—try
running them directly in
your terminal.

**Your APIs may not be considered
“mission critical” like Stripe. But
your API design is a contract with
your API consumers.**



Mehdi Medjaoui

@medjawii



Following

.@Stripe maintains 50 different versions of its API to respect its "write once, run everywhere" business promise. #TechnicalDebtTradeOff

**Your API design is composed of
the capabilities (or “skills”)
you offer to developers**

API Skills == “I want to...”



Save Gmail attachments to Dropbox as original ...

Use This



Share new Tweets from a Twitter list in Slack

Use This

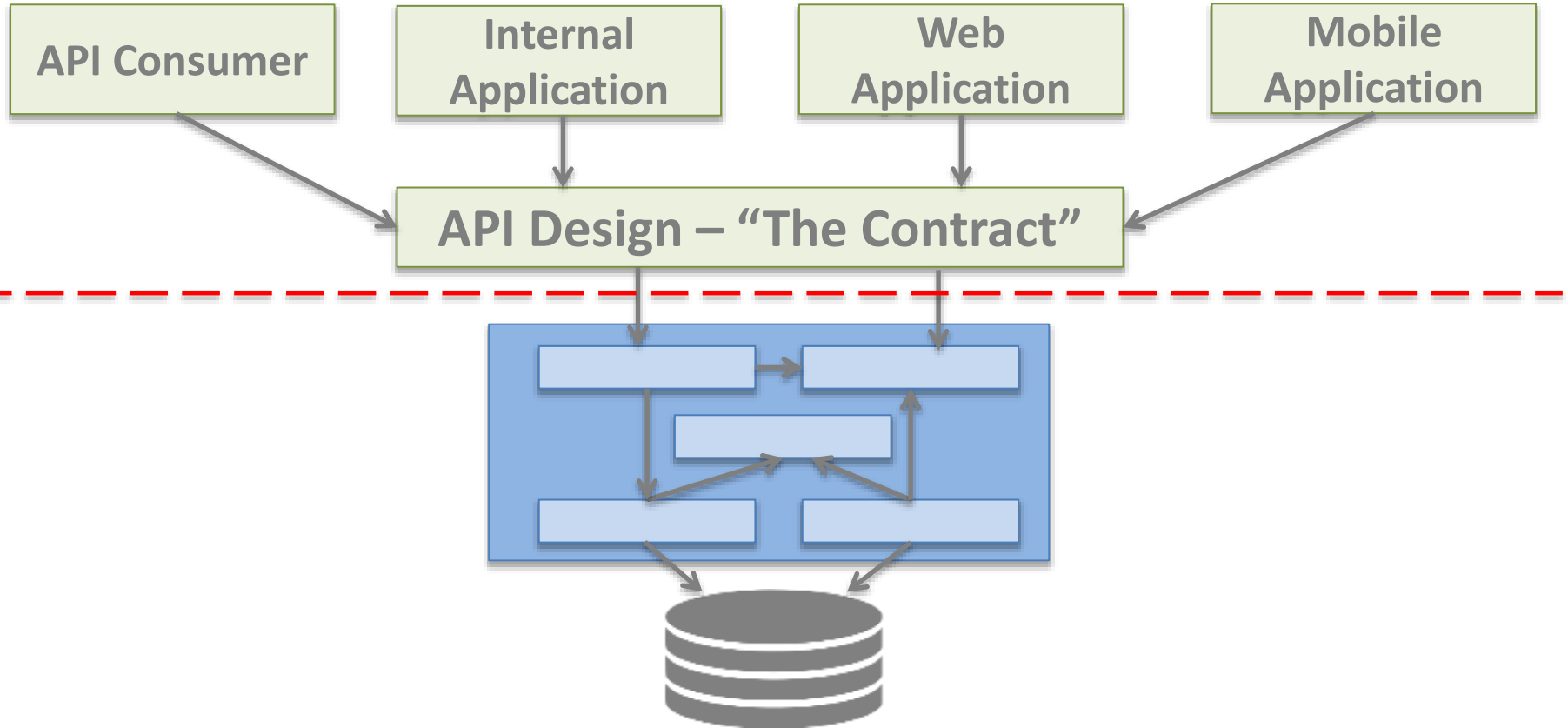


Create Trello card weekly

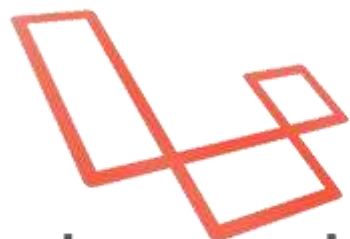
Use This

**What skills do your
APIs need to offer?**

Outside-In API Design Approach



API Design is an architectural concern that combines business, product design, and software engineering



laravel

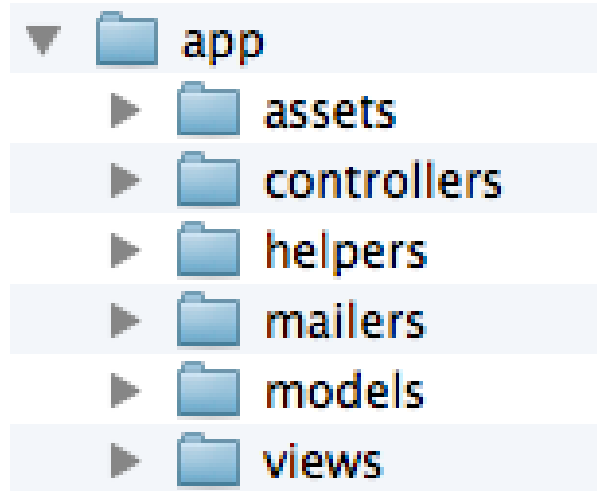


django

express

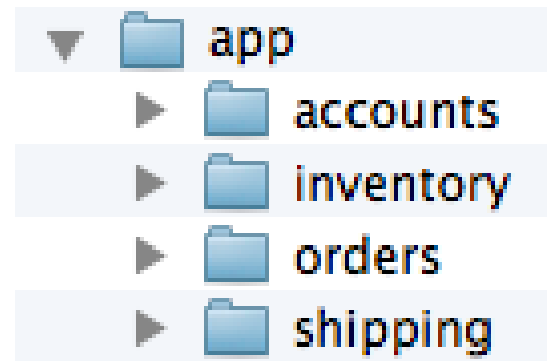
Lack of Modularization = Regret

Function-Based



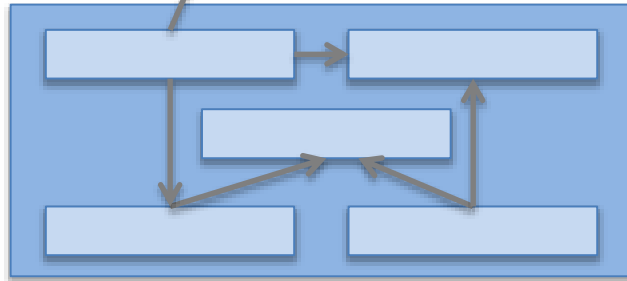
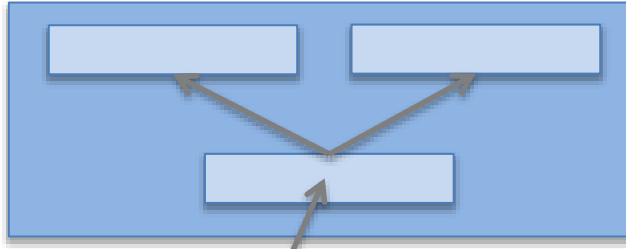
VS

Module-Based



High Cohesion + Loose Coupling

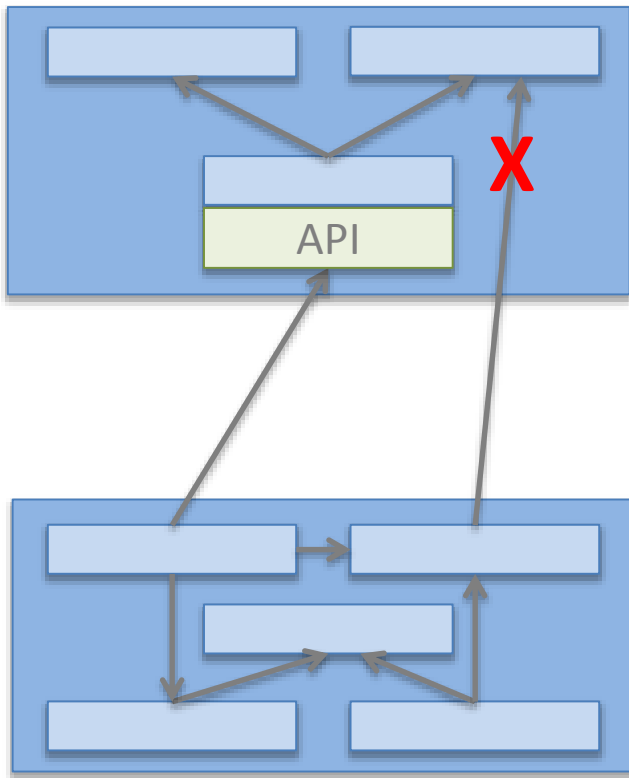
High cohesion
internally for common
functionality



Loose coupling
externally across
modules

API-Centric Software Design

Public APIs
encourage
loose coupling
between modules



Scoping rules
restrict access
outside of modules

Systems Design

System

Subsystem

Module

Module

Module

Module

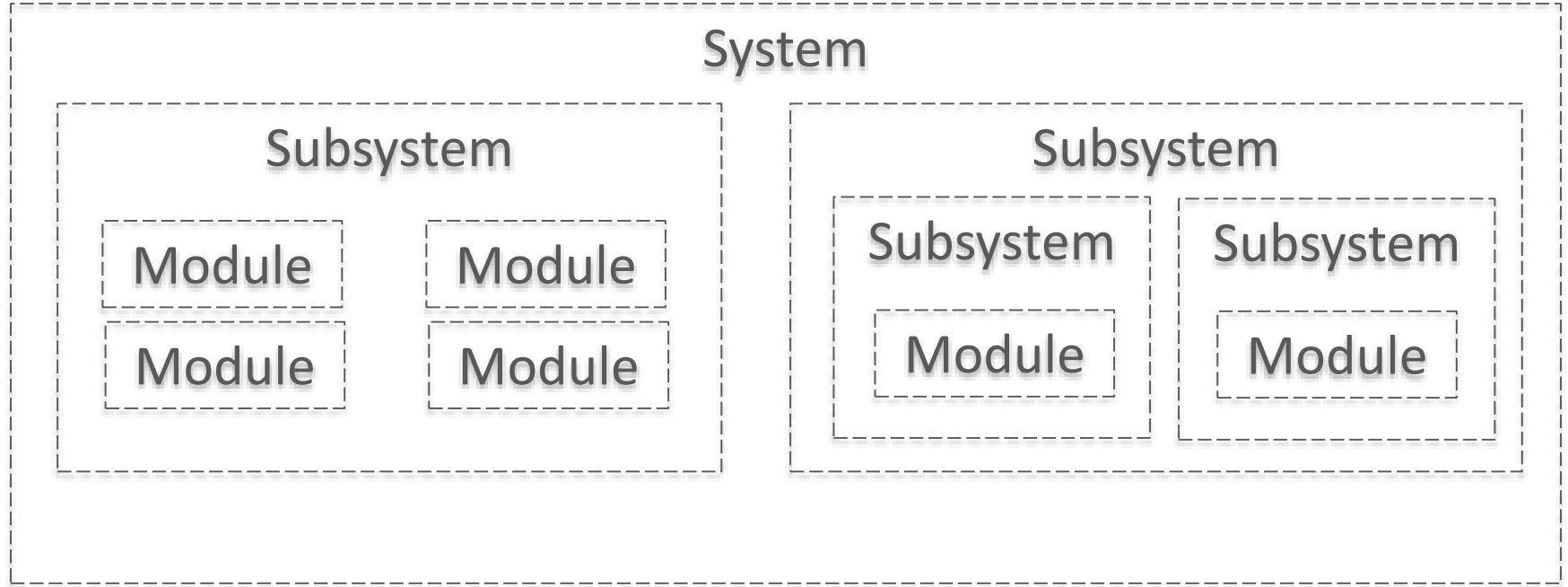
Subsystem

Subsystem

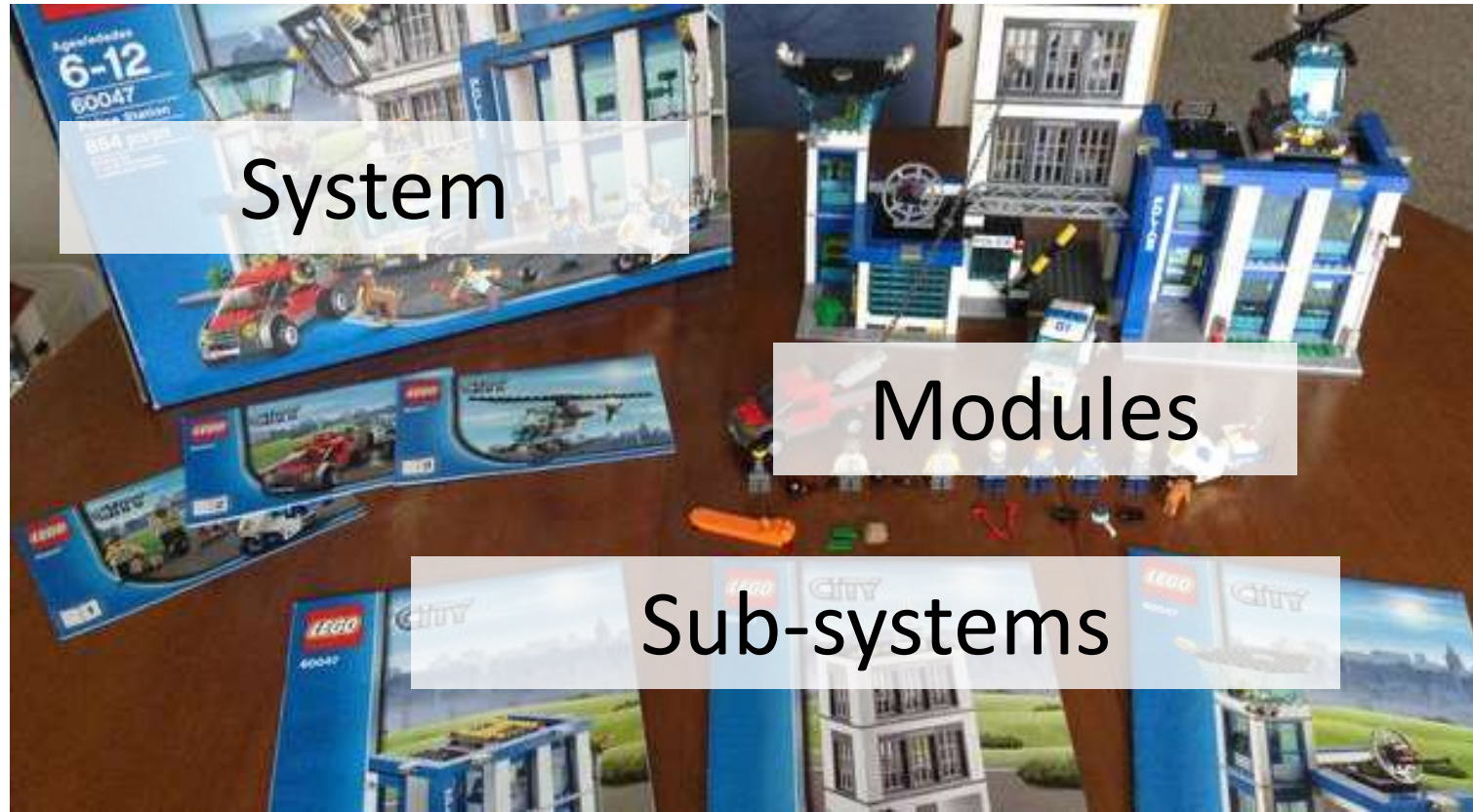
Subsystem

Module

Module



LEGO as Modular System Design

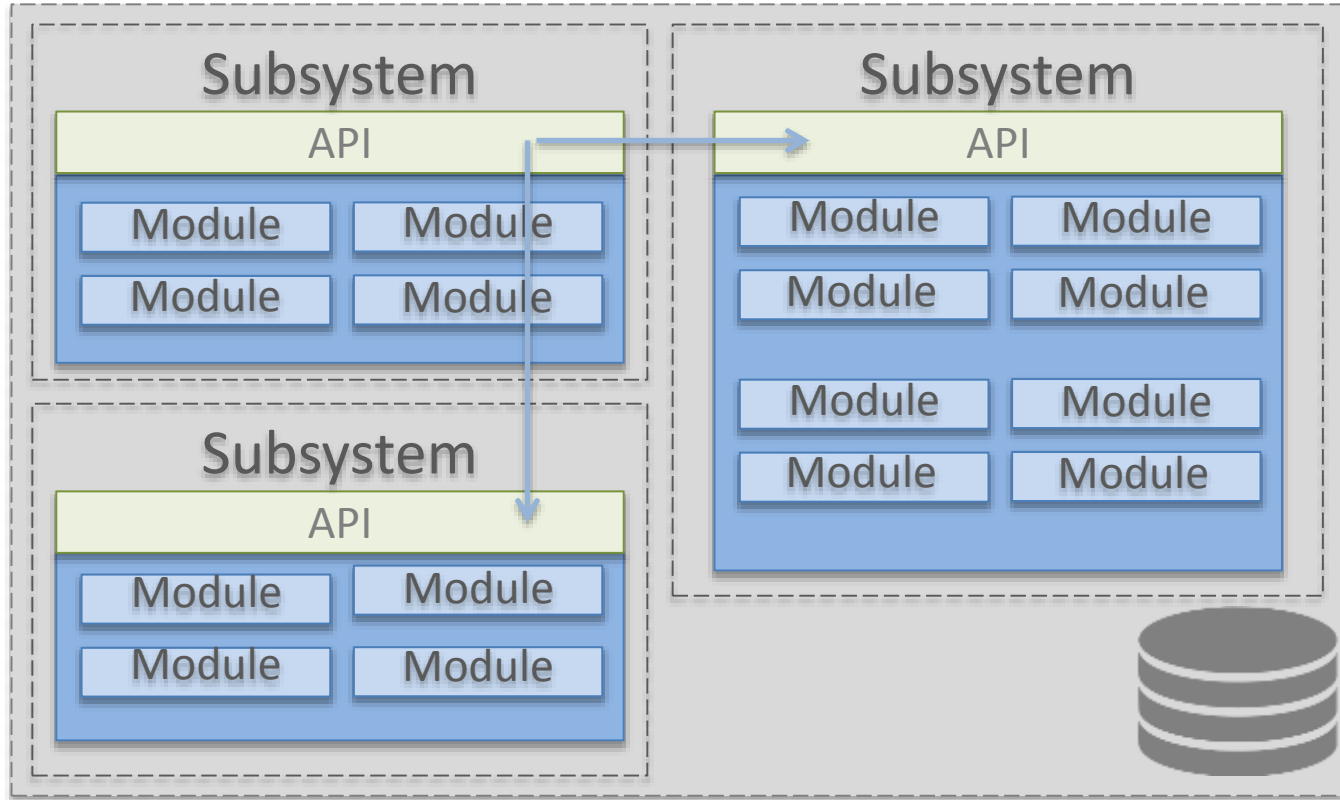


System

Modules

Sub-systems

System/Solution



**Domain-Driven Design helps
with identifying context
boundaries for complex APIs**

Domain Driven Design (DDD)

- ◆ Maps domain concepts into software
- ◆ Heavy domain expert involvement
- ◆ Common vocabulary (“Ubiquitous Language”)
- ◆ Boundary-driven (“Bounded Context”)
- ◆ Applies learning over time

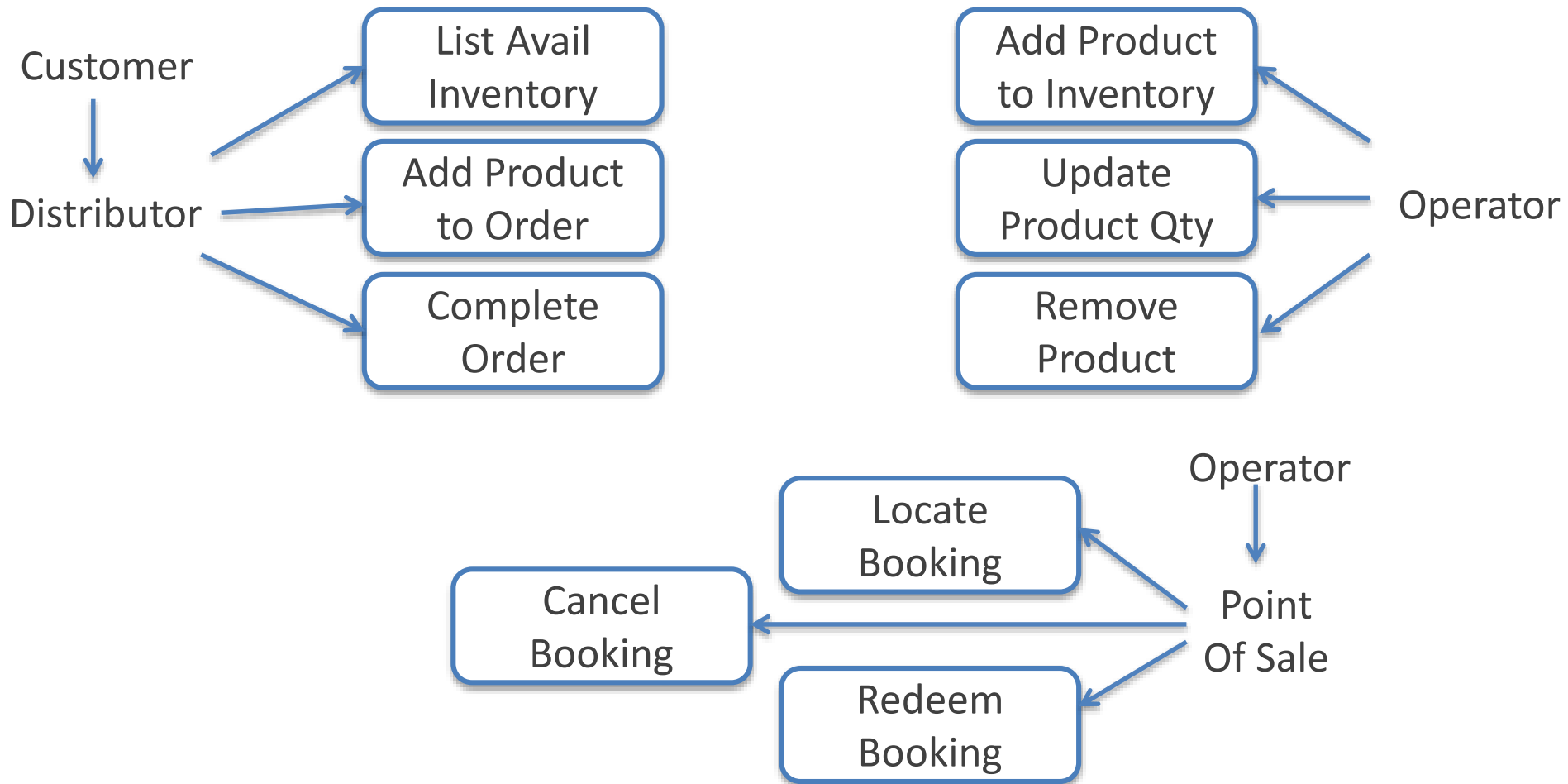
**“I need to multiply two numbers
together. So, I am building
a microservice architecture.”
- almost everyone today**

Microservice Architecture

- ◆ Loosely-coupled, service-oriented architecture
- ◆ Apply bounded context to limit cognitive load
- ◆ Independently deployable via automation
- ◆ Enable replaceability and experimentation
- ◆ Encourage composability of the business
- ◆ Best for larger teams

**Applying Systems Design
and Domain-Driven Design
to find API boundaries
and resources**

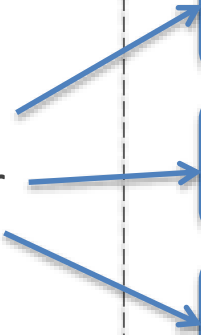
| Activity | Step | Participant(s) | Description |
|-------------|-----------------------|----------------|---|
| Place Order | List Items | Customer | List/search for store items |
| Place Order | Add Item To Cart | Customer | Add an item to the customer's cart |
| Place Order | Remove Item From Cart | Customer | Remove an item from the customer's cart |
| Place Order | Clear Cart | Customer | Remove all items from the customer's cart |
| Place Order | View Cart | Customer | View the current cart and total |
| Place Order | Checkout | Customer | Create an order from the contents of the cart |



Customer



Distributor



Order Mgmt

List Avail
Inventory

Add Product
to Order

Complete
Order

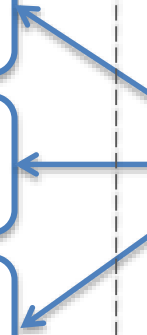
Inventory Mgmt

Add Product
to Inventory

Update
Product Qty

Remove
Product

Operator



Fulfillment

Locate
Booking

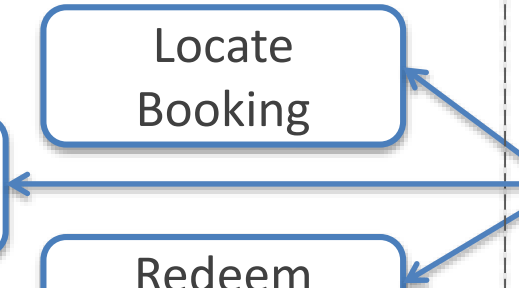
Cancel
Booking

Redeem
Booking

Operator



Point
Of Sale

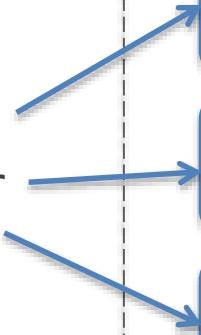


**Each subsystem has an API that
exposes one or more endpoints**

Customer



Distributor



Order API

List Avail
Inventory

Add Product
to Order

Complete
Order

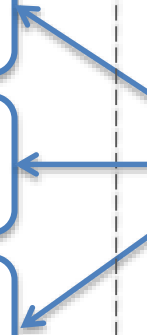
Inventory API

Add Product
to Inventory

Update
Product Qty

Remove
Product

Operator



Fulfillment API

Locate
Booking

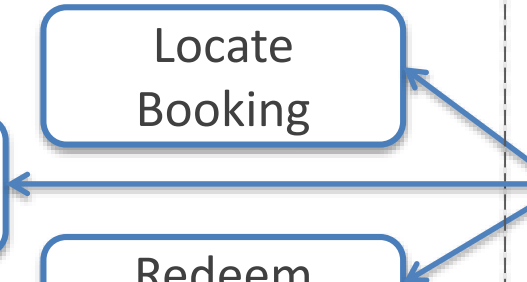
Cancel
Booking

Redeem
Booking

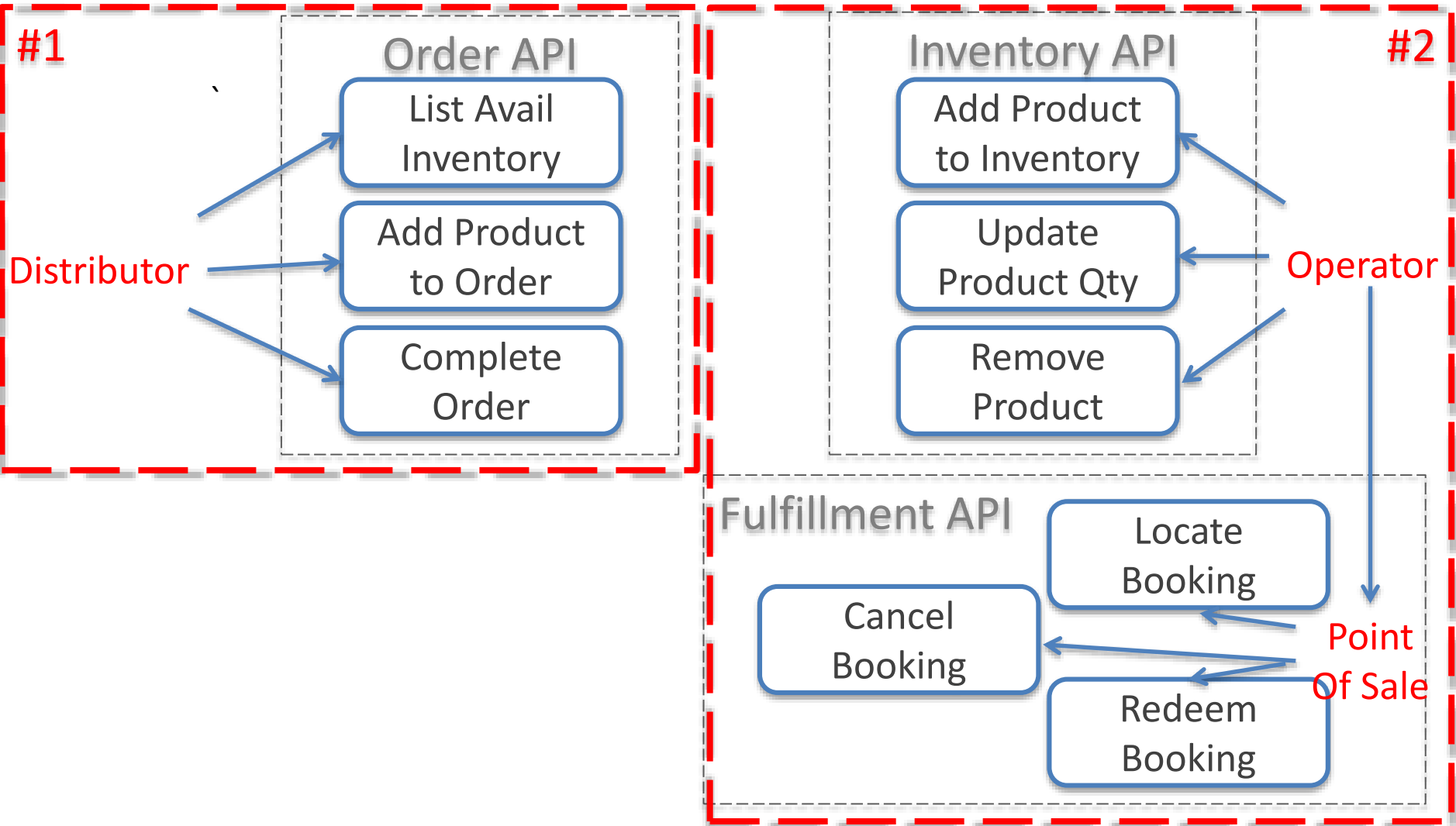
Operator



Point
Of Sale



**Identifying product opportunities
becomes easier when we have
boundaries around the APIs**



For every API, ask:
“What resources will it offer?”

**Use domain-driven design to
find business entities, relations,
state transitions, and events**

Order API

List Avail
Inventory

Add Product
to Order

Complete
Order

Inventory API

Add Product
to Inventory

Update
Product Qty

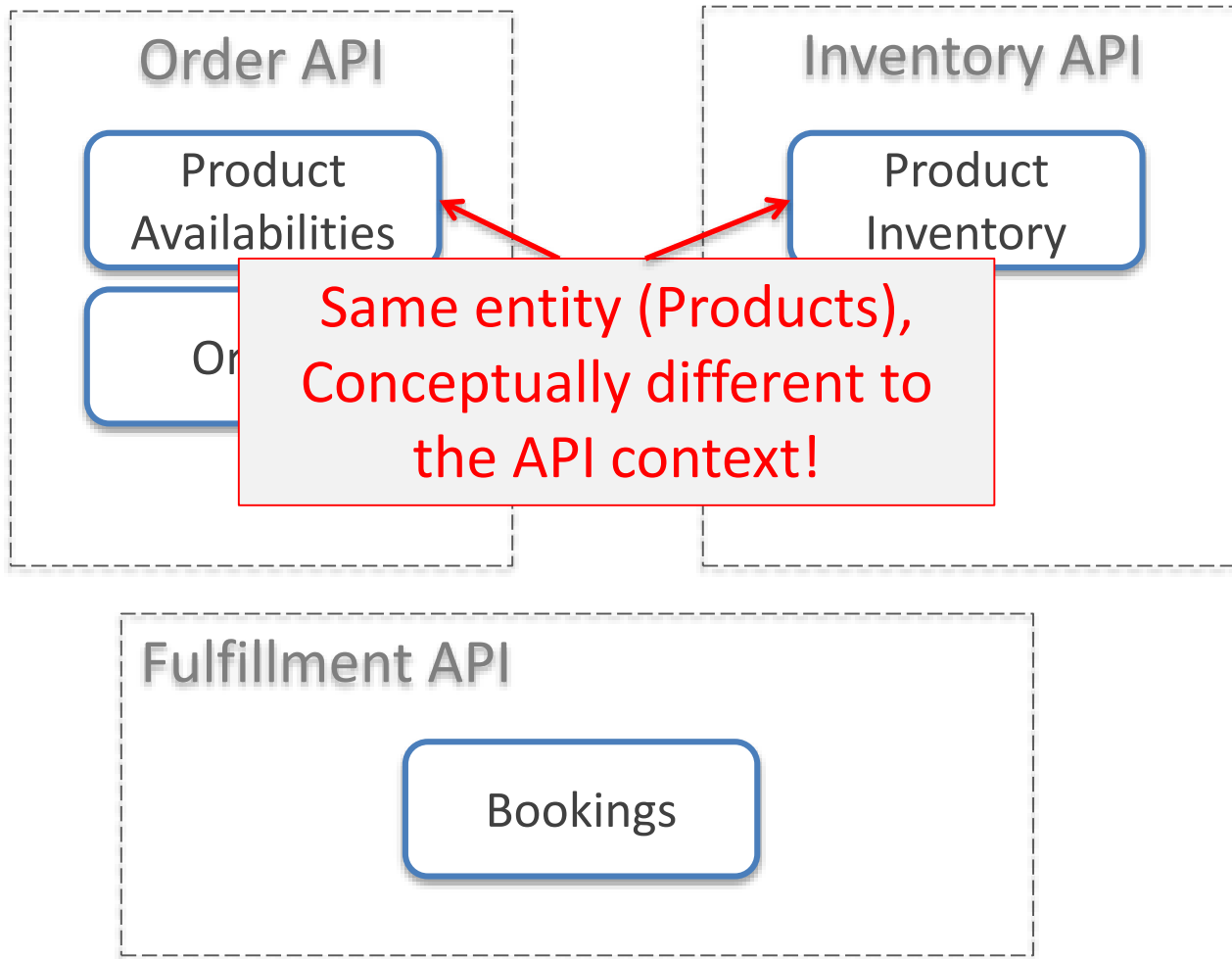
Remove
Product

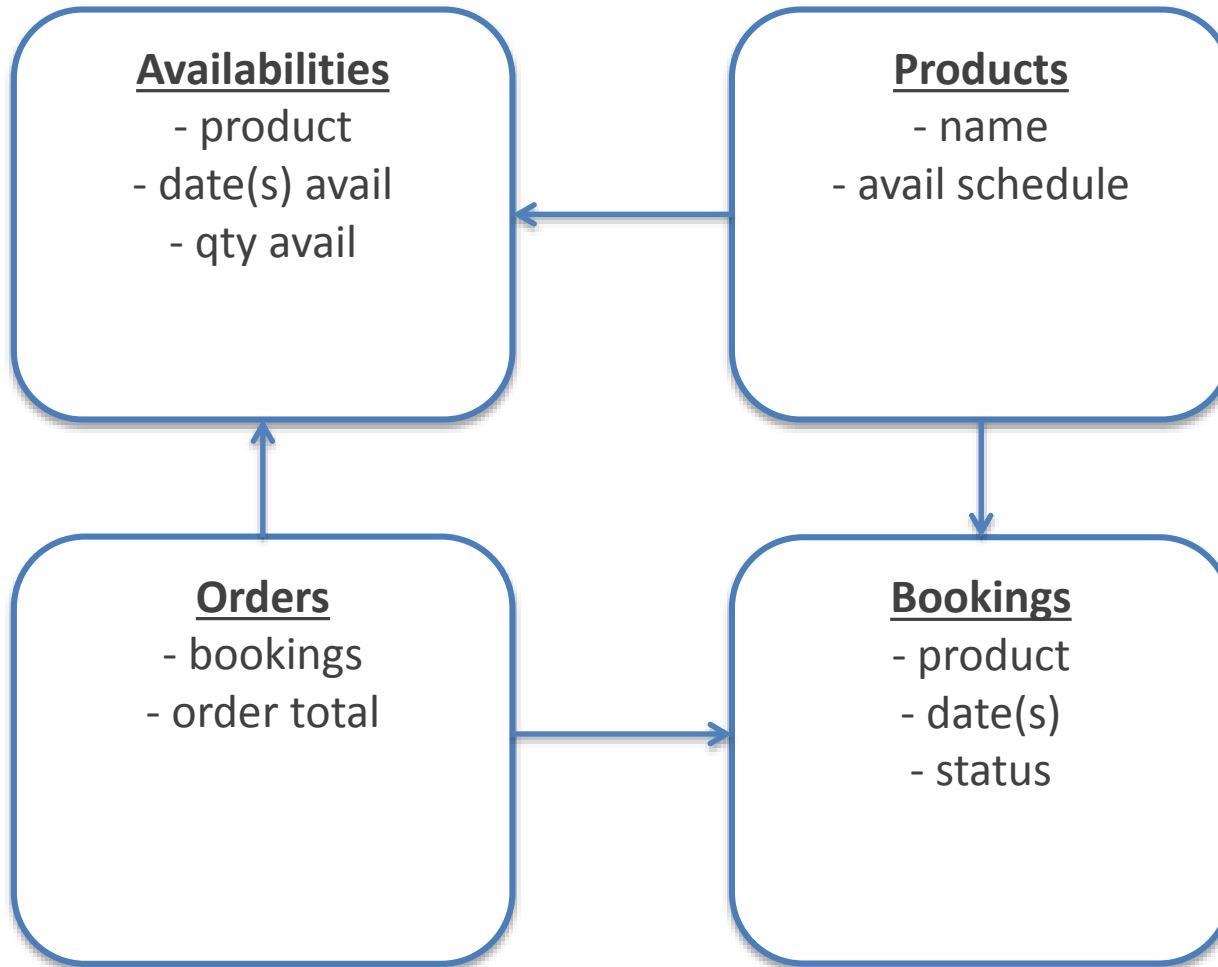
Fulfillment API

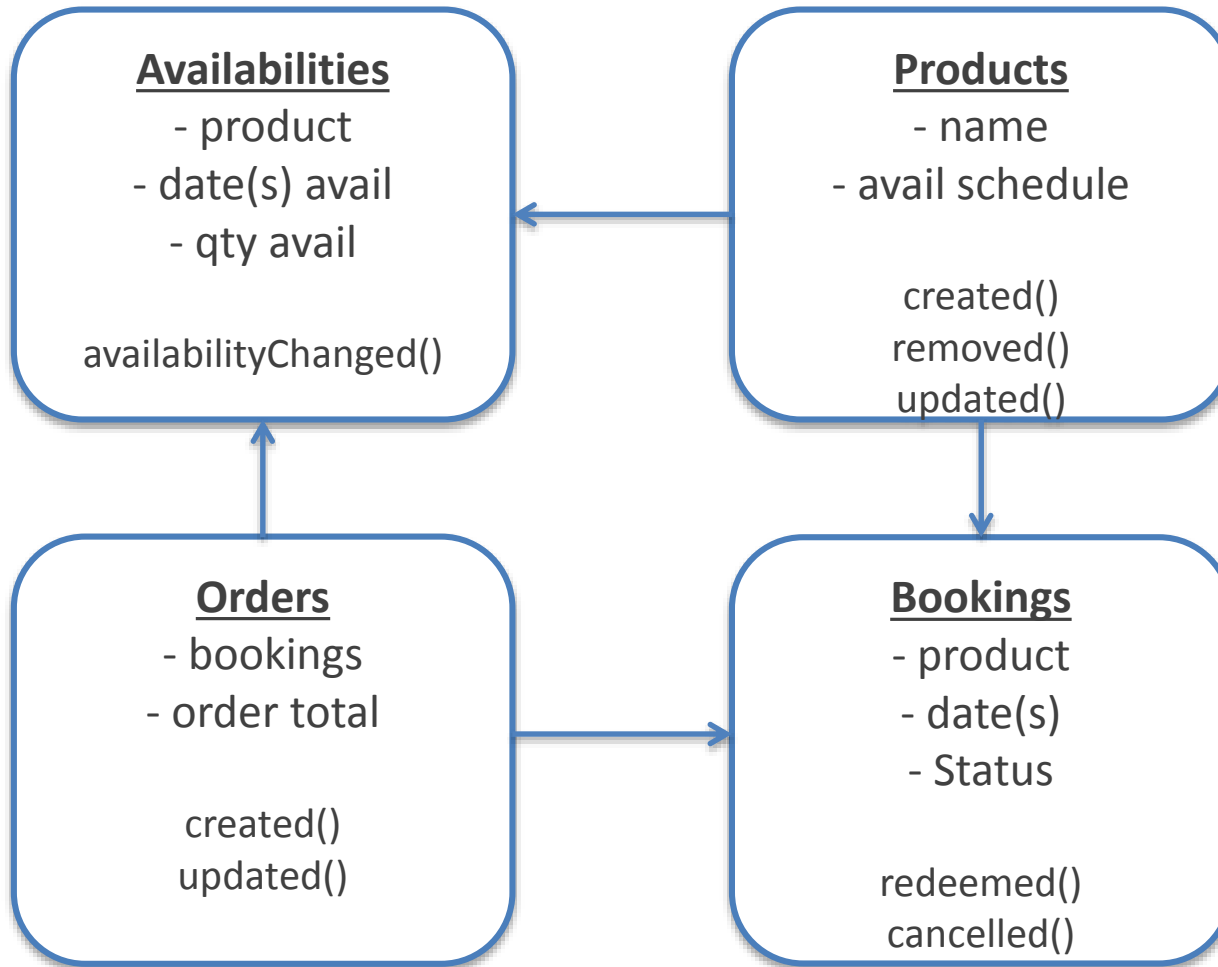
Cancel
Booking

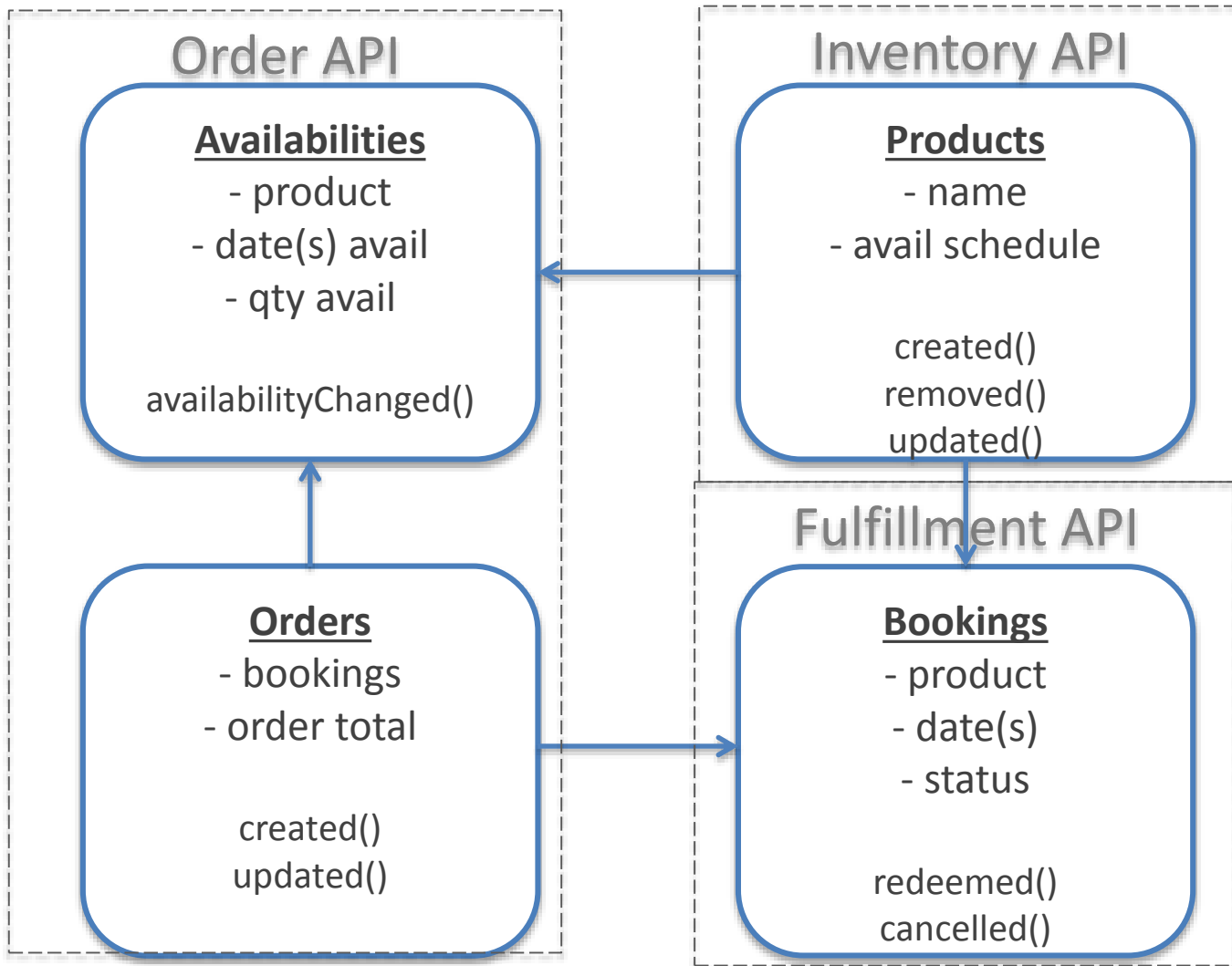
Locate
Booking

Redeem
Booking









Map Resources to the API

| Endpoint | Description | Response Codes |
|-----------------------------|--|---|
| GET /cart | List details about the cart, including total cost and all items/quantities in our cart | 200 OK |
| POST /cart/items | Add an item to the cart with a specific quantity | 201 Created 401 Unauthorized 400 Bad Request (item already in cart) |
| PUT /cart/items/{itemId} | Update an item quantity already in the cart | 200 OK 401 Unauthorized 404 Not found (item not found in cart) |
| DELETE /cart/items | Clear the contents of the cart | 204 No Content 401 Unauthorized |
| DELETE /cart/items/{itemId} | Remove an item from the cart | 204 No Content 401 Unauthorized 404 Not found (item not found in cart) |

API composability using microservices

**Public APIs target durability
through external contracts.**

**Microservice APIs target evolution through
learning and experimentation**

Orders HTTP API

Availabilities Service

Orders Service

Inventory HTTP API

Products Service

Fulfillment HTTP API

Bookings Service



Payment Service

Refund Service

Orders HTTP API

Availabilities Service

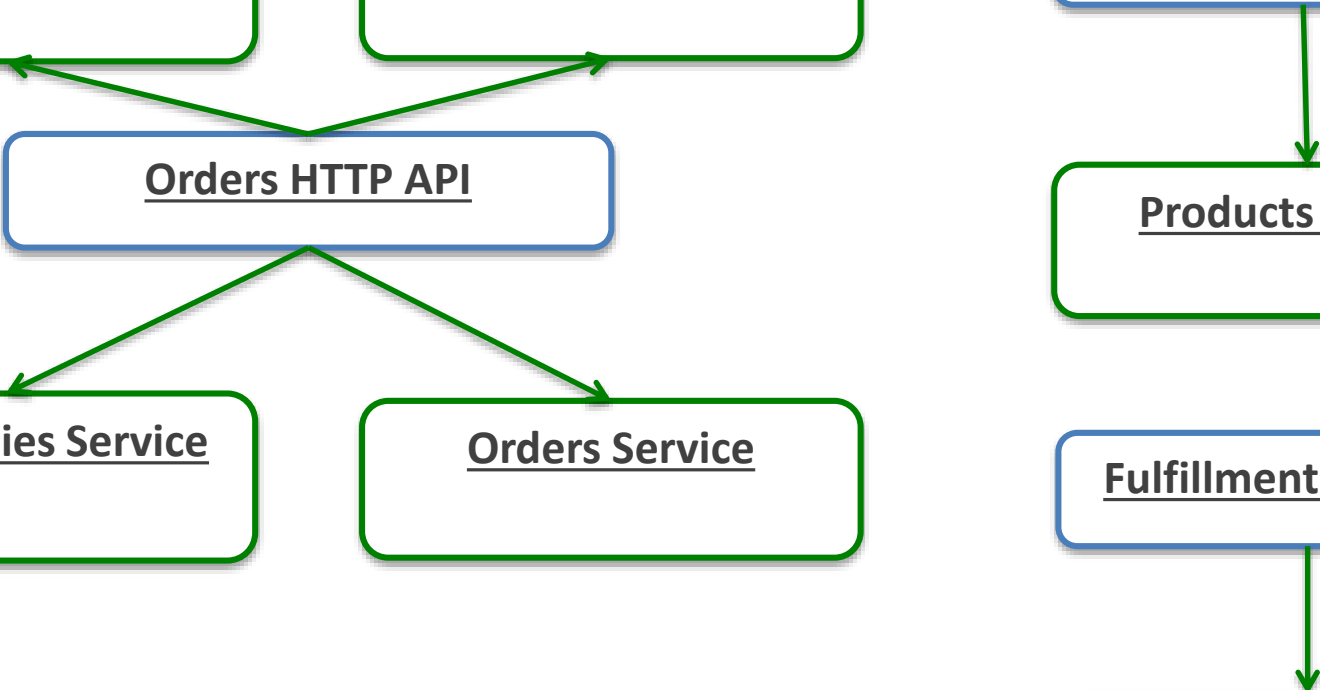
Orders Service

Inventory HTTP API

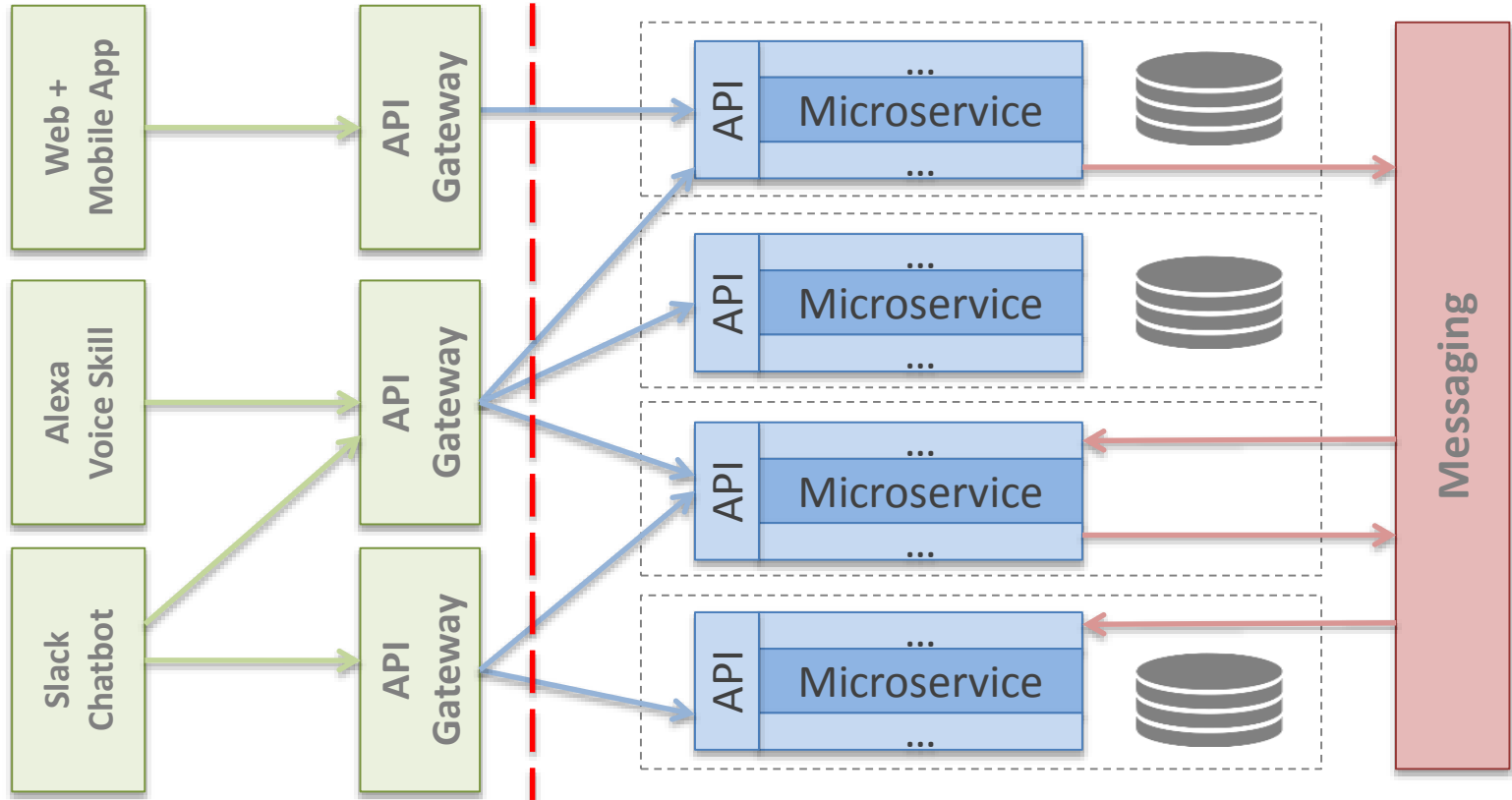
Products Service

Fulfillment HTTP API

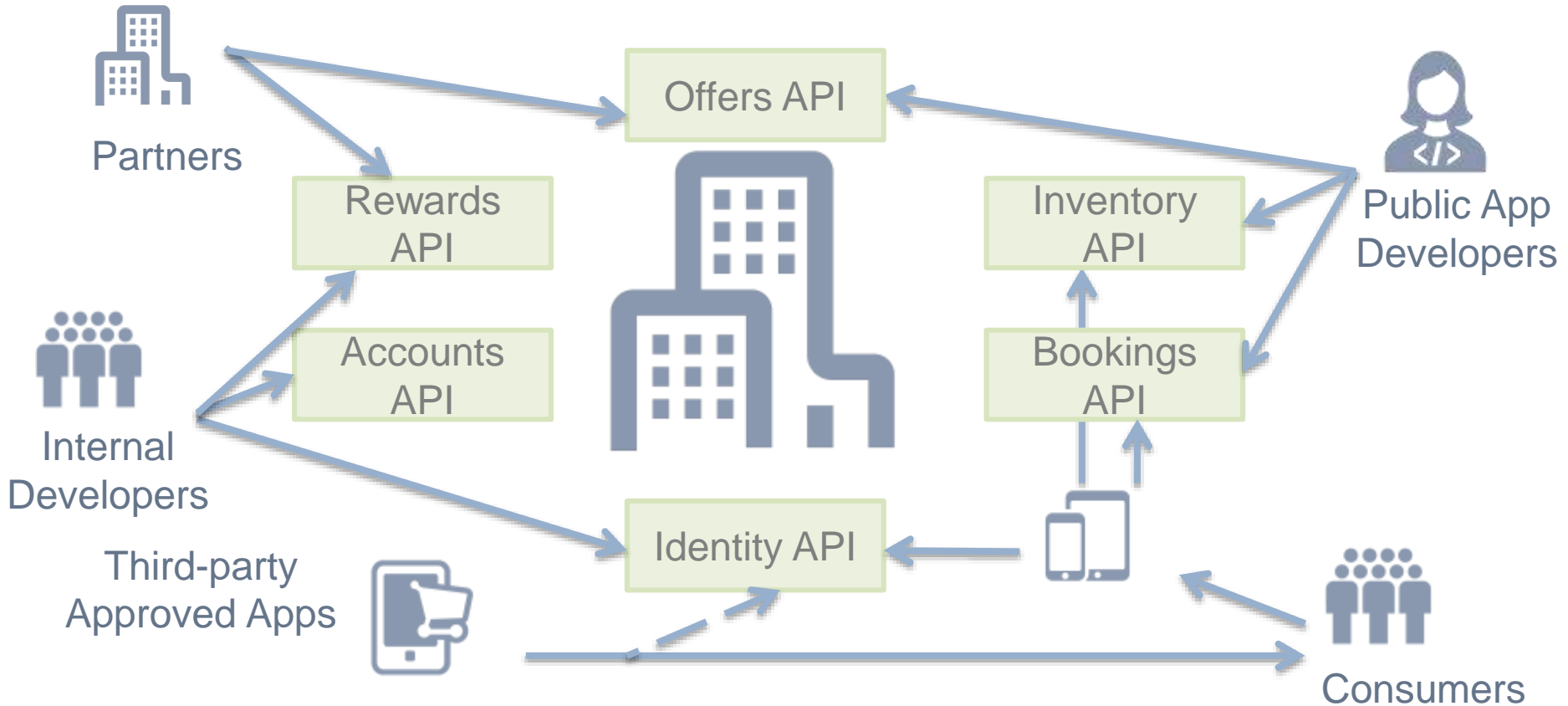
Bookings Service



Microservice Architecture



The Composable World of APIs



Thank you

James Higginbotham

@launchany

<http://bit.ly/api-skills>

