
# Arrested Development

## The awkward adolescence of a microservices-based application

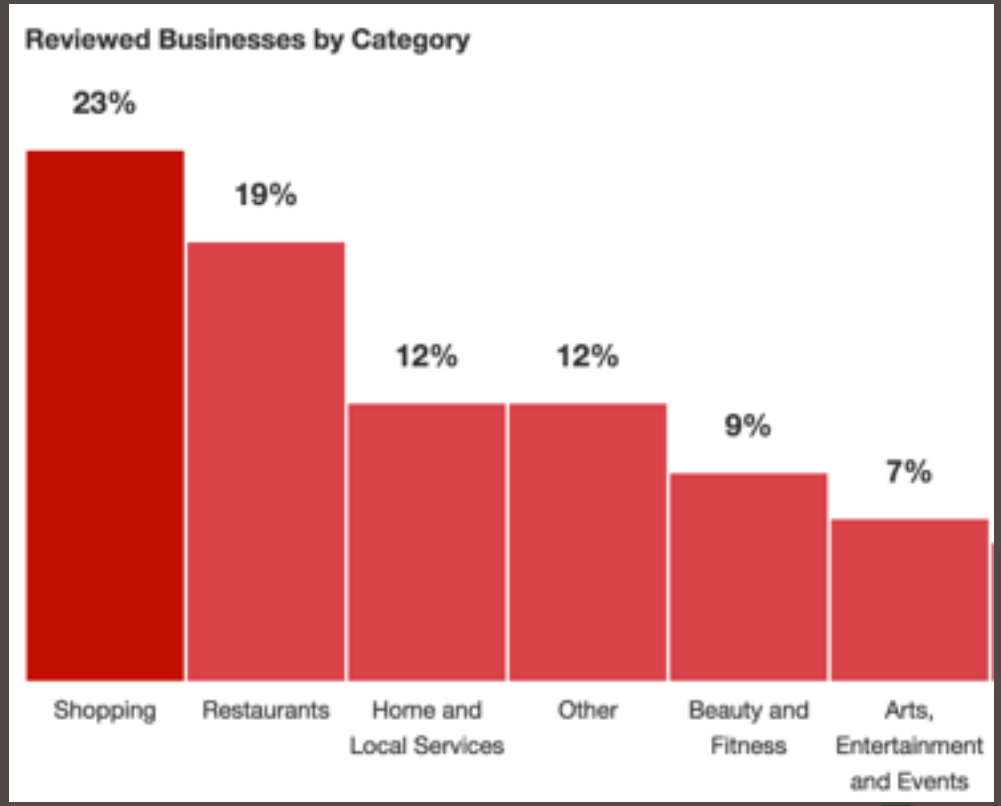
Europython 2015

Scott Triglia


The Company

77M reviews

142M monthly unique users



Scott Triglia
@scott_triglia

Your Speaker

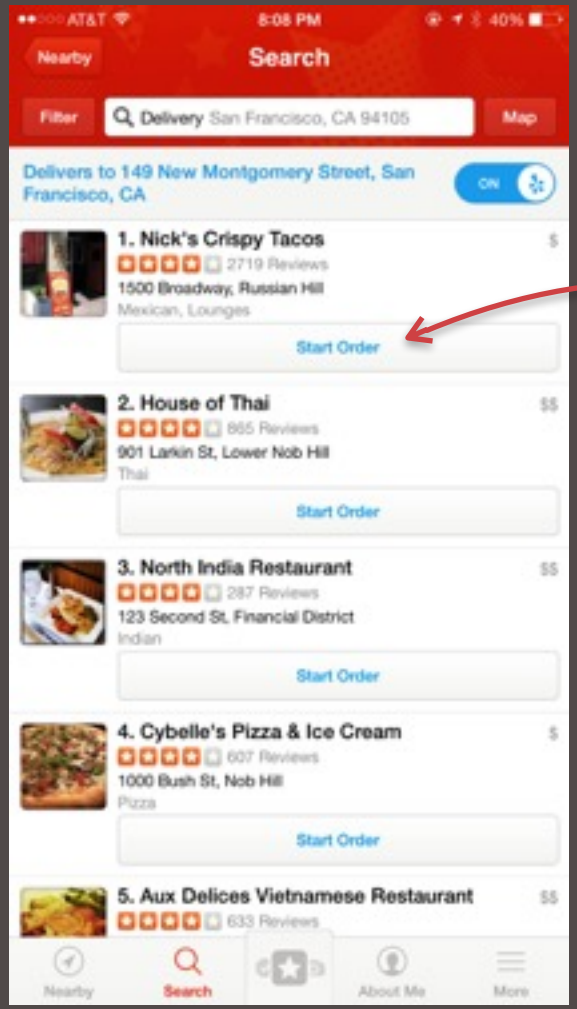
4 years with Yelp
Search, ML, Services

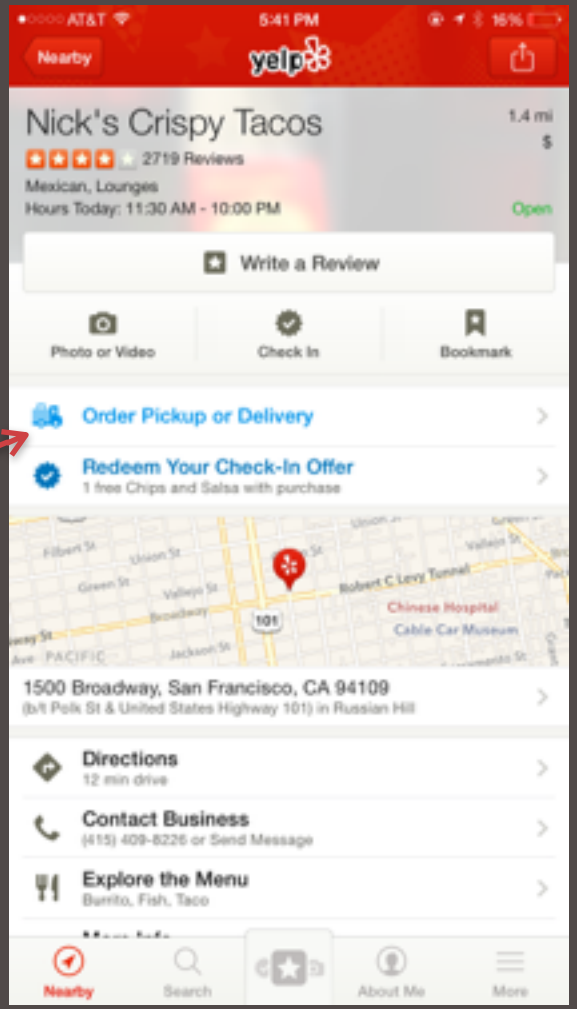
Yelp
Transaction
Platform

The Product

Yelp
Transaction
Platform
(or just “Platform”)

The Product


AT&T
8:08 PM
40%
Nearby
Search
Filter
Delivery San Francisco, CA 94105
Map
Delivers to 149 New Montgomery Street, San Francisco, CA
ON
1. Nick's Crispy Tacos
$
2719 Reviews
1500 Broadway, Russian Hill
Mexican, Lounges
Start Order
2. House of Thai
$$
865 Reviews
901 Larkin St, Lower Nob Hill
Thai
Start Order
3. North India Restaurant
$$
287 Reviews
123 Second St, Financial District
Indian
Start Order
4. Cybelle's Pizza & Ice Cream
$
607 Reviews
1000 Bush St, Nob Hill
Pizza
Start Order
5. Aux Delices Vietnamese Restaurant
$$
633 Reviews
Nearby
Search
About Me
More


AT&T
5:41 PM
16%
Nearby
yelp
Nick's Crispy Tacos
1.4 mi
$
2719 Reviews
Mexican, Lounges
Hours Today: 11:30 AM - 10:00 PM
Open
Write a Review
Photo or Video
Check In
Bookmark
Order Pickup or Delivery
Redeem Your Check-In Offer
1 free Chips and Salsa with purchase
Robert C Levy Tunnel
Chinese Hospital
Cable Car Museum
101
1500 Broadway, San Francisco, CA 94109
(b/t Polk St & United States Highway 101) in Russian Hill
Directions
12 min drive
Contact Business
(415) 409-8226 or Send Message
Explore the Menu
Burrito, Fish, Taco
Nearby
Search
About Me
More

AT&T 8:08 PM 40%

Cancel

## Make Your Selection

### Nick's Crispy Tacos

Delivery $5.00 · Min $9.00

Estimated Delivery Time 45-75 min

ORDERING BY EAT24

**Full Menu**

Menu hours: 10:30 AM to 9:00 PM

**What's Good**

**Pescado Taco** $4.95

Baja style fried fish in a corn tortilla with salsa, lime mayonnaise, cilantro, cabbage and onions.

**Carne Asada Taco** $3.95

Grilled steak in corn tortilla with pico de gallo and pinto beans.

**Chips and Salsa** $1.25

**Nachos** $7.95

Our house made chips topped with a traditional cheese sauce, black beans,

AT&T 8:17 PM 40%

Cancel

Make Your Selection

Powered by
SHOPTIQUES.COM

BAG (0)

BACK

CATHEDRAL MAXI DRESS, $149.00

COLOR

SIZE XS S M What's my size?

ADD TO BAG

DESCRIPTION

Lose fitting midi length scarf maxi dress with adjustable straps. Belt it or leave it loose for a great country vibe. Size small measures 51" at longest length.

AT&T 8:23 PM 38%

Cancel

Make Your Selection

Le Meridien San Francisco

Fri, 8/7 - Sat, 8/8 (1 night) | 2 guests edit

powered by hipmunk via Orbitz

1 Choose Room

1K bed:classic non-smoking room:55 inch led tv :...
Non-refundable
view details
$346

1K bed:classic accessible:visual fire alarm :visual ...
Non-refundable
view details
$346

1K bed:superior non-smoking:55 inch led tv :wi-fi i...
Non-refundable
view details
$355

view all 58 rooms

2 Guest Information

Same as billing information

Different from billing information

First Name

Last Name

AT&T 5:48 PM 13%

Cancel

## Summary

Pickup
Nick's Crispy Tacos

Use coupon

**Carne Asada Taco** $3.95 x 2 = **$7.90**

| | |
|---|---|
| Subtotal | $7.90 |
| Tax | $0.69 |
| SF Healthy Mandate | $0.32 |
| **Total** | **$8.91** |

Edit

### Your Contact Information

**Phone Number** 5554379125

### Payment Information

VISA Use VISA ****-6444 >

By proceeding, you agree to our Terms of Service and Privacy Policy. We'll send your transaction to a partner company to fulfill.

**Complete Purchase**

# Microservices

That Hot Trend

“...an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms...”

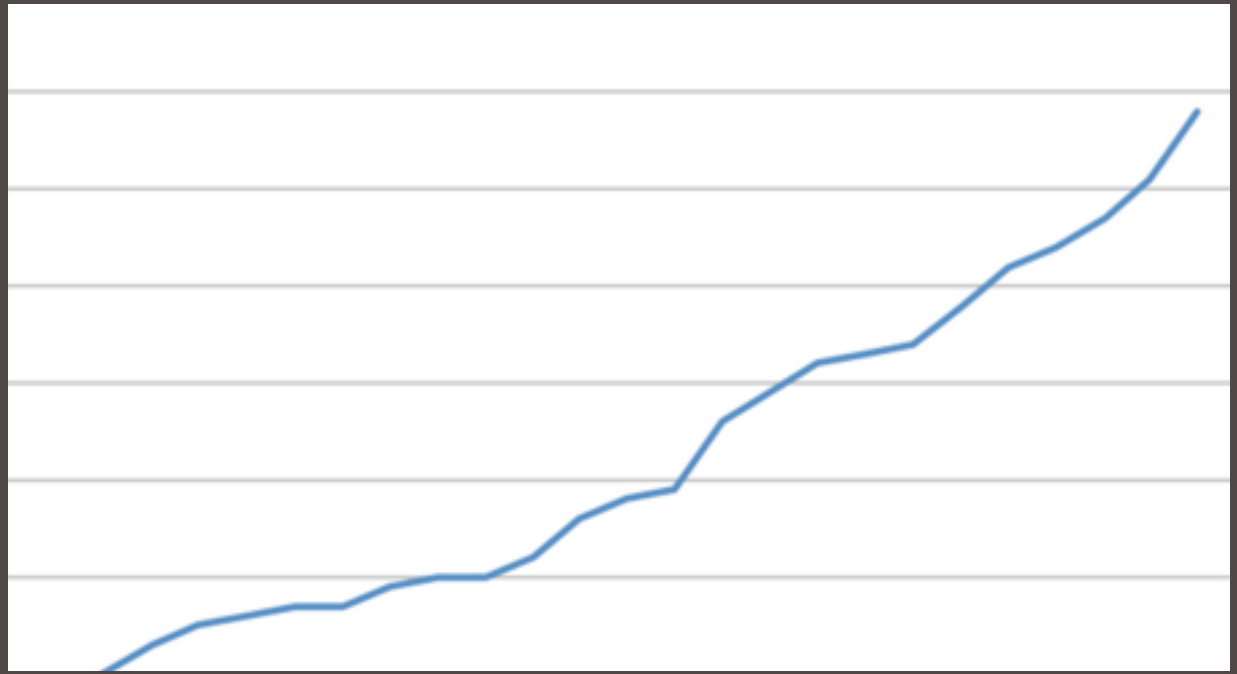
http://martinfowler.com/articles/microservices.html

# Monolithic python code resisted decoupling

Monolithic python code catered to the lowest common denominator

Monolithic python code was anti-agile

Services



Time

Pinterest Gingerbread House

Pinterest Gingerbread House

# API complexity increases

# coupling rises

# interactions get murky

# process does not scale

# So what’s an engineer to do?

- Decoupling
- Defining
- Understanding Production
- Staying Agile

- Decoupling
- Defining
- Understanding Production
- Staying Agile

# Old boring problem

## Monolithic spaghetti code

# Solution: microservices!

# New exciting problem

## how to share concepts across services

# New exciting problem

# distributed tech debt

# service_type

# service_type

What product does your business provide and how do they provide it?

# service_type

pickup

delivery

# service_type

- booking_at_customer
- pickup
- booking_at_business
- delivery

# service_type

booking_at_customer

hotel_reservation

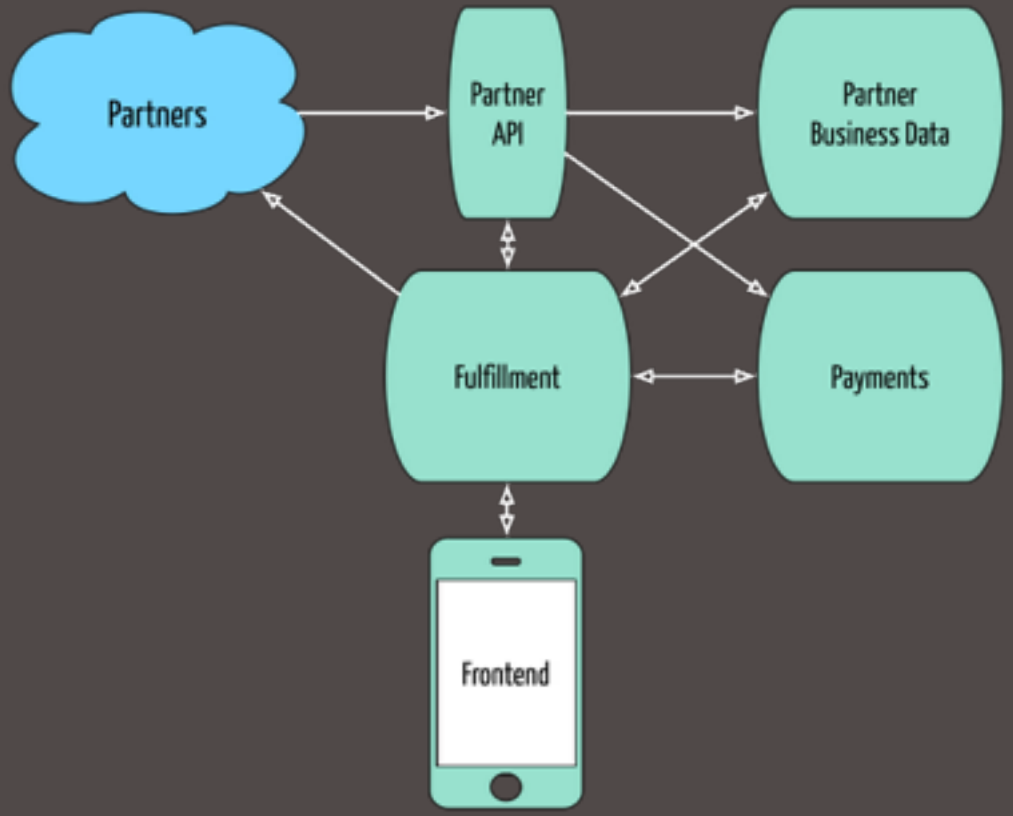
goods_at_business

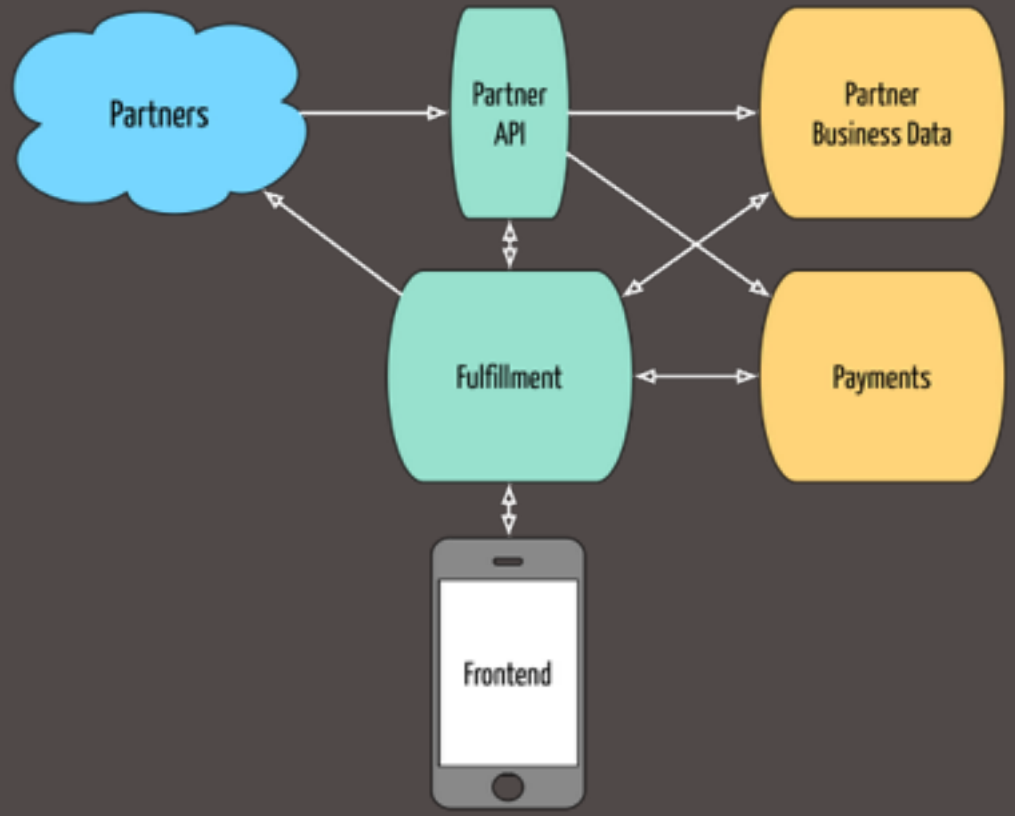
pickup

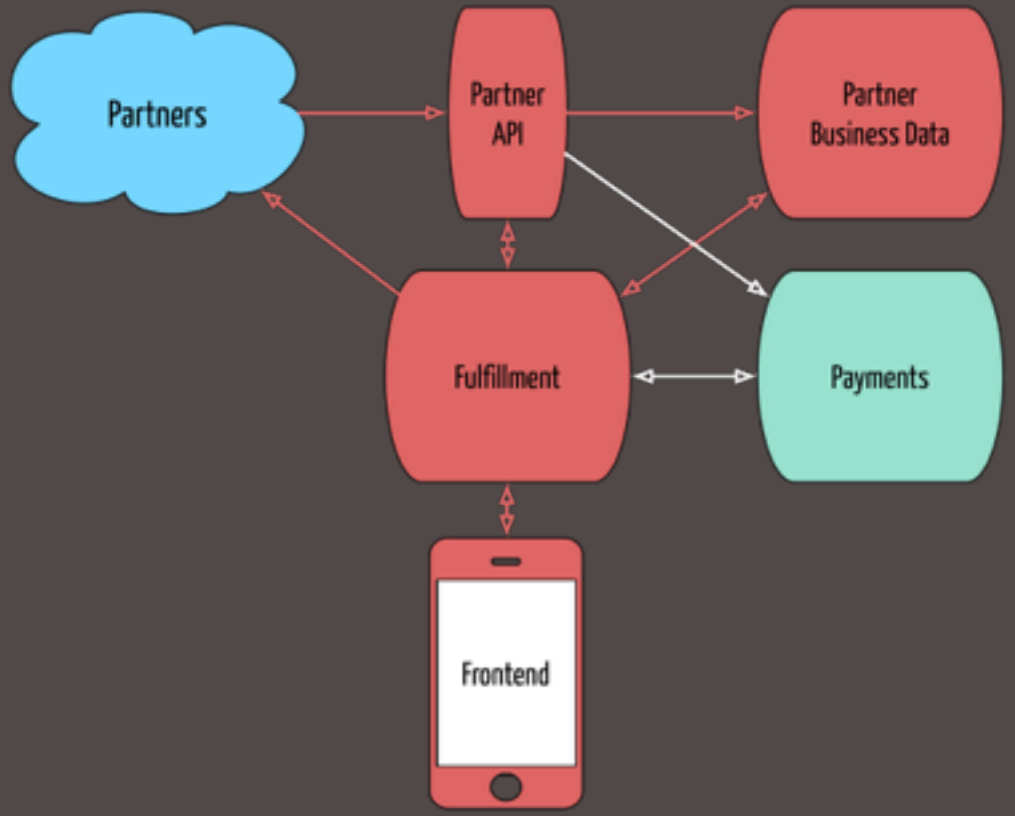
booking_at_business

goods_at_customer

delivery


Partners
Partner API
Partner Business Data
Fulfillment
Payments
Frontend

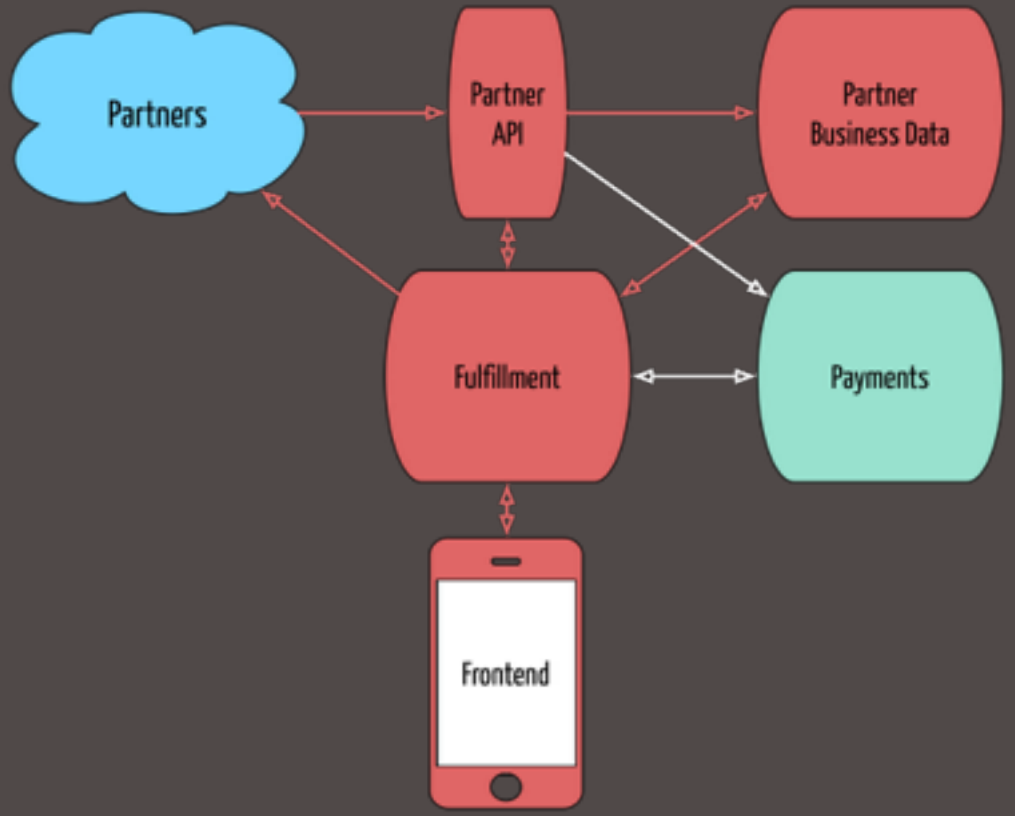

Partners
Partner API
Partner Business Data
Fulfillment
Payments
Frontend


Partners
Partner API
Partner Business Data
Fulfillment
Payments
Frontend

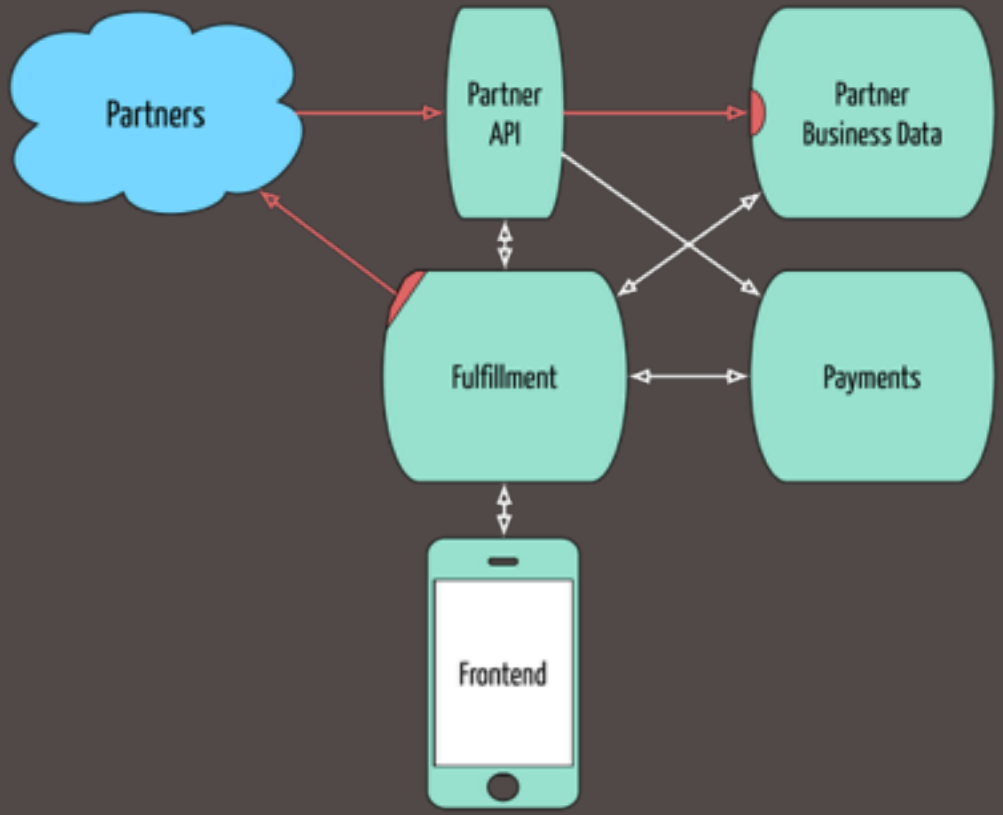
Confusing

Pervasive

Convenient, but not designed


Partners
Partner API
Partner Business Data
Fulfillment
Payments
Frontend

Draw boundaries, introduce domain-specific concepts tied to functionality


Partners
Partner API
Partner Business Data
Fulfillment
Payments
Frontend

# Lessons

Interfaces are the sum of APIs, shared libraries, and the data that flows through them

Sacrificing DRYness can be the best choice for overall design

Service interfaces are a great opportunity to intentionally decouple systems

- Decoupling
- Defining
- Understanding Production
- Staying Agile

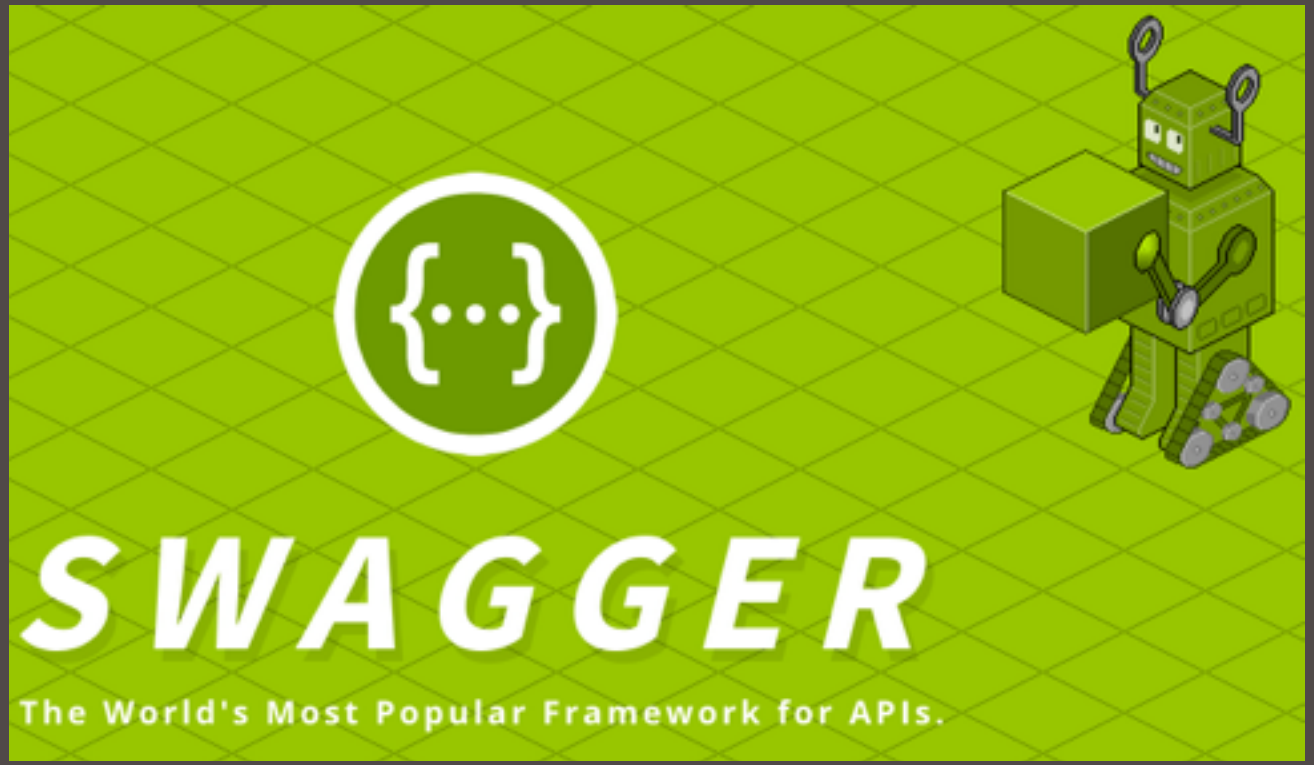
Have you ever needed to understand a system and been told go read the source?

What about a system which only validates half its interface?

Coming from a python monolith, strong interfaces were quite rare

```
def checkout(order, price, **kwargs):
    “““Process an order.”””

    validate_order(order)
    charge_credit_card(order.user, price)
    notify_user(order, **kwargs)
```


SWAGGER
The World's Most Popular Framework for APIs.

```yaml
# will be prefixed to all paths
basePath: /v1
produces:
  - application/json
paths:
  /products:
    get:
      summary: Product Types
      description: |
        The Products endpoint returns information about the *Uber* products
        offered at a given location. The response includes the display name
        and other details about each product, and lists the products in the
        proper display order.
      parameters:
        - name: latitude
          in: query
          description: Latitude component of location.
          required: true
          type: number
          format: double
        - name: longitude
          in: query
          description: Longitude component of location.
          required: true
          type: number
          format: double
      tags:
        - Products
```

swagger

http://petstore.swagger.io/v2/swagger.json

api_key

Explore

# Swagger Petstore

This is a sample server Petstore server. You can find out more about Swagger at http://swagger.io or on irc.freenode.net, #swagger. For this sample, you can use the api key `special-key` to test the authorization filters.

Find out more about Swagger

http://swagger.io
Contact the developer
Apache 2.0

## pet : Everything about your Pets

Show/Hide | List Operations | Expand Operations

| | | |
|---|---|---|
| POST | /pet | Add a new pet to the store |
| PUT | /pet | Update an existing pet |
| GET | /pet/findByStatus | Finds Pets by status |
| GET | /pet/findByTags | Finds Pets by tags |
| DELETE | /pet/{petId} | Deletes a pet |
| GET | /pet/{petId} | Find pet by ID |

# Client side - Yelp/bravado

```
from bravado.client import SwaggerClient

client = SwaggerClient.from_url(
    “www.myservice.com/swagger.json”
)

pet = client.pet.getPetById(petId=42).result()
```

# Server side - striglia/pyramid_swagger

```
# In your Pyramid webapp.py
config.include(‘pyramid_swagger')
```

# Lessons

# Interfaces should be intentional

# Interfaces should be explicit

Find the mechanical things which don’t scale and automate them mercilessly

- Decoupling
- Defining
- Understanding Production
- Staying Agile

Real customer bug report:

"We're seeing 504s talking to the /user_info API"

Ancient times:

Use logic and whatever logs happen to exist

# Better:

Log all incoming API requests to any service

(spam via Flickr; CC by 2.0)

Best:

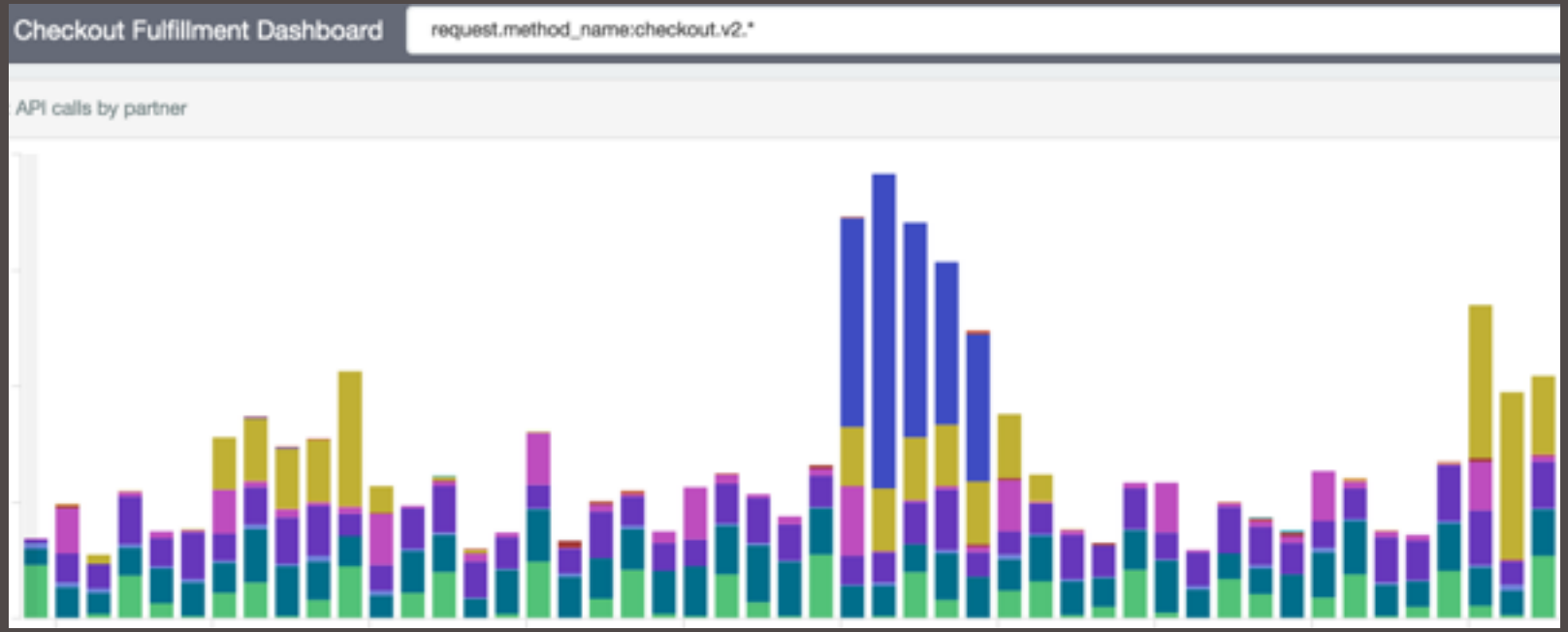
Every service has a detailed access/error log and tooling to examine them

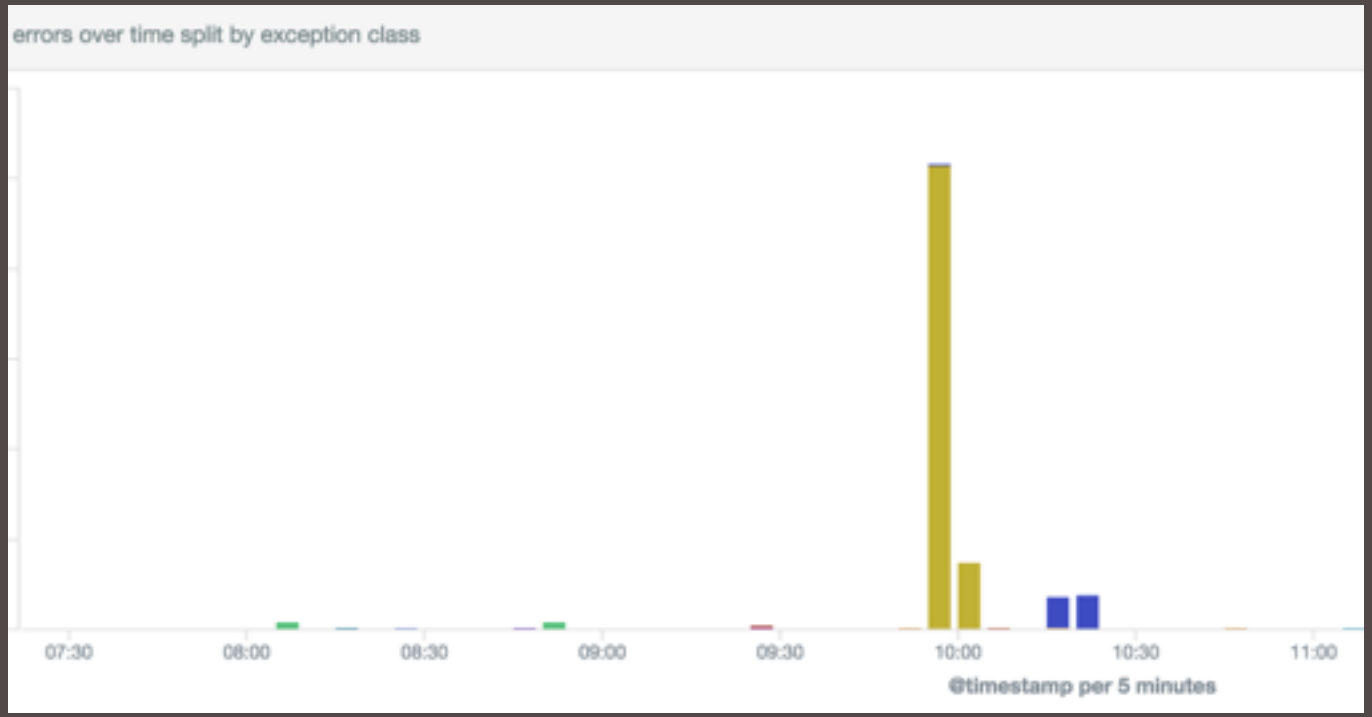
elasticsearch.

kibana

elasticsearch.

kibana


Checkout Fulfillment Dashboard
request.method_name:checkout.v2.*
API calls by partner

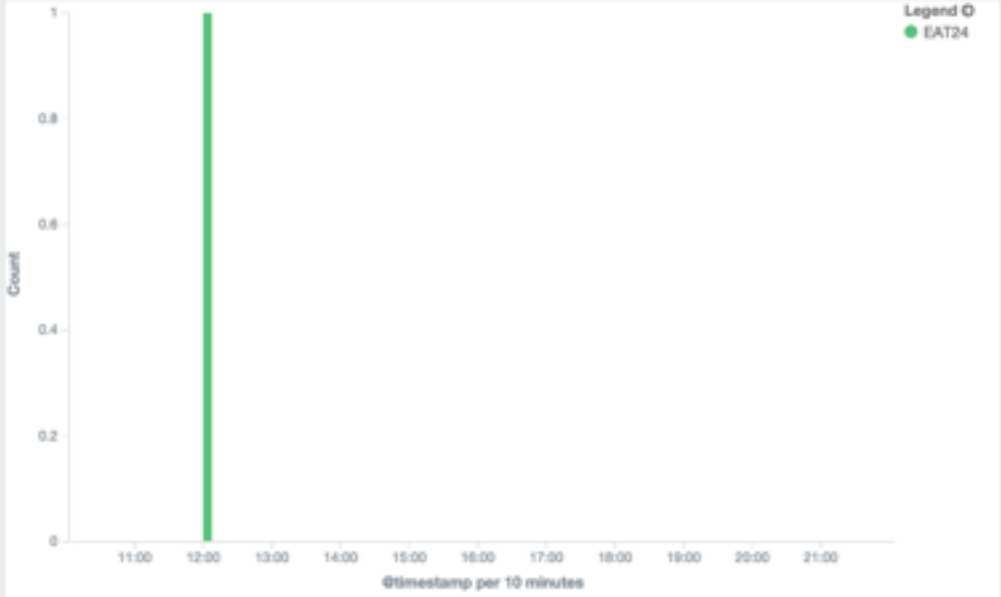

errors over time split by exception class
07:30
08:00
08:30
09:00
09:30
10:00
10:30
11:00
@timestamp per 5 minutes

# So what about that customer with the mystery 504?

YTP: Checkout Fulfillment Dashboard

request.method_name.raw:checkout.v2.orders.fulfillment_user_info AND NOT response.status_code:200

YTP: API calls by partner

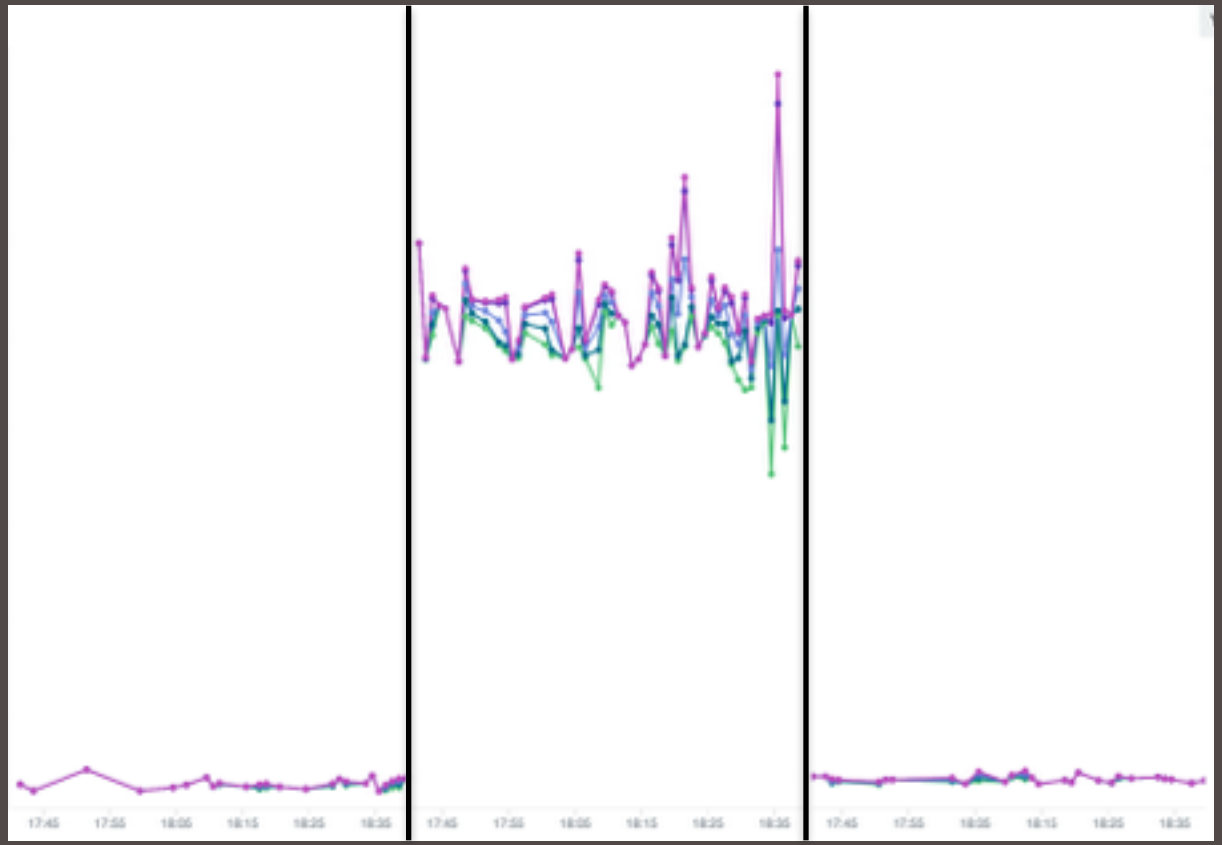



YTP: All CF requests

| Time | request.method_name | request.params.yelp_order_id |
|---|---|---|
| July 12th 2015, 12:02:23.821 | checkout.v2.orders.fulfillment_user_info | eb8b228a026960981bc5fb038760139c |

2.5 s

0.15 s



Realistically:

Don’t require the customer to report issues in the first place

Yelp / elastalert

Watch 44

Easy & Flexible Alerting With ElasticSearch https://elastalert.readthedocs.org

300 commits | 17 branches | 34 releases | 15 contributors

branch: master elastalert / +

Merge pull request #158 from sophicware/feature/smtp_ssl ...

Qmando authored 6 days ago latest commit e74eea9e6f

```
es_host: elasticsearch-hostname
es_port: 14900
index: logstash-errors-%G.%V

type: frequency
num_events: 20
timeframe:
  minutes: 2

alert:
  - "modules.sensu_alert.SensuAlerter"
sensu:
  team: platform
  tip: "This alert indicates a large number of errors across the Platform
product. See <link to Kibana> for details."
  page: true
  status: 2  # CRITICAL
```

```
es_host: elasticsearch-hostname
es_port: 14900
index: logstash-errors-%G.%V

type: frequency
num_events: 20
timeframe:
  minutes: 2

alert:
  - "modules.sensu_alert.SensuAlerter"
sensu:
  team: platform
  tip: "This alert indicates a large number of errors across the Platform
product. See <link to Kibana> for details."
  page: true
  status: 2  # CRITICAL
```

```
es_host: elasticsearch-hostname
es_port: 14900
index: logstash-errors-%G.%V

type: frequency
num_events: 20
timeframe:
  minutes: 2

alert:
  - "modules.sensu_alert.SensuAlerter"
sensu:
  team: platform
  tip: "This alert indicates a large number of errors across
the Platform product. See <link to Kibana> for details."
  page: true
  status: 2  # CRITICAL
```

# Lessons

Logging is a superpower. Use it ~~wisely~~ constantly.

But raw data is not enough! Visualize and monitor actively.

These approaches make a world of difference:

- Incident response from days to minutes
- Investigations from ∞ to minutes

- Decoupling
- Defining
- Understanding Production
- Staying Agile

Uncomfortable conversation:

"Customers had their orders interrupted. How are you preventing it going forward?"

# Understandable response:

# “Deploy more carefully”

# Understandable response:

# “Expand oncall”

How do we ensure the team stays agile as our services grow in complexity?

Pain point:

The testing environment is {broken, flaky, not like prod}


docker

# Pain point:

Tests passed but production broke

Production monitoring is the natural extension of excellent pre-deploy testing.

Yelp / elastalert

Watch 44

Easy & Flexible Alerting With ElasticSearch https://elastalert.readthedocs.org

300 commits | 17 branches | 34 releases | 15 contributors

branch: master elastalert / +

Merge pull request #158 from sophicware/feature/smtp_ssl ...

Qmando authored 6 days ago latest commit e74eea9e6f

Pain point:

No clue how much time we spend fixing production issues

# Pain point:

Tough to argue what changes will make things more robust

**Outage Started On** *

mm/dd/yyyy, --:-- --

Example: 03/05/2013 11:30 AM

**Outage Finished On** *

mm/dd/yyyy, --:-- --

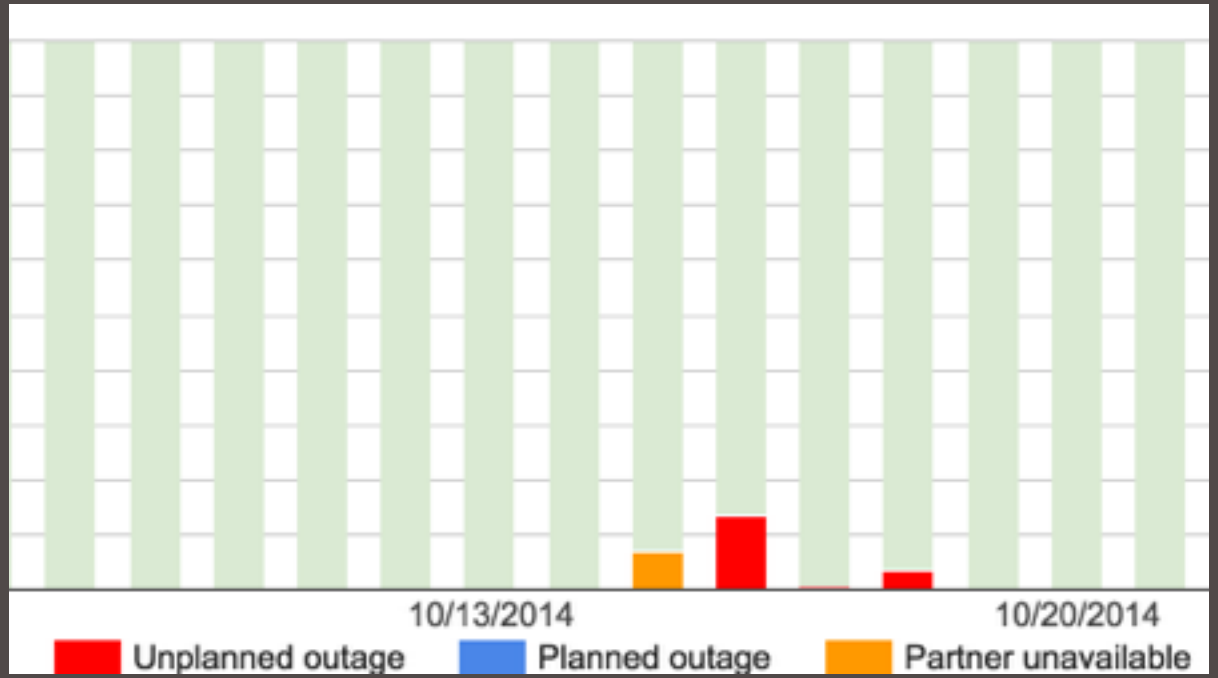
Example: 03/05/2013 11:30 AM

**Did this impact a specific partner, or all of them?** *

**Was this a planned outage?** *

- Unplanned
- Planned

**What is the nature of the outage?** *


10/13/2014
10/20/2014
Unplanned outage
Planned outage
Partner unavailable

And as with everything else, this must eventually be automated

# Lessons

Networks of services are fundamentally harder to test. Prepare accordingly.

Failure will happen. Focus on both identifying and recovering quickly.

Staying agile is easy if your application rarely fails and recovers automatically

# Wrap Up

# Know your roots

# Be explicit

# Measure everything

# Scale via automation

Yelp/bravado

striglia/pyramid_swagger

Yelp/elastalert

http://engineeringblog.yelp.com/2015/03/using-services-to-break-down-monoliths.html

# Our accumulated wisdom

Yelp/service-principles

# Questions?

@scott_triglia

scott.triglia@gmail.com