



DELIVERING WITH MICROSERVICES

HOW TO ITERATE TOWARDS SOPHISTICATION

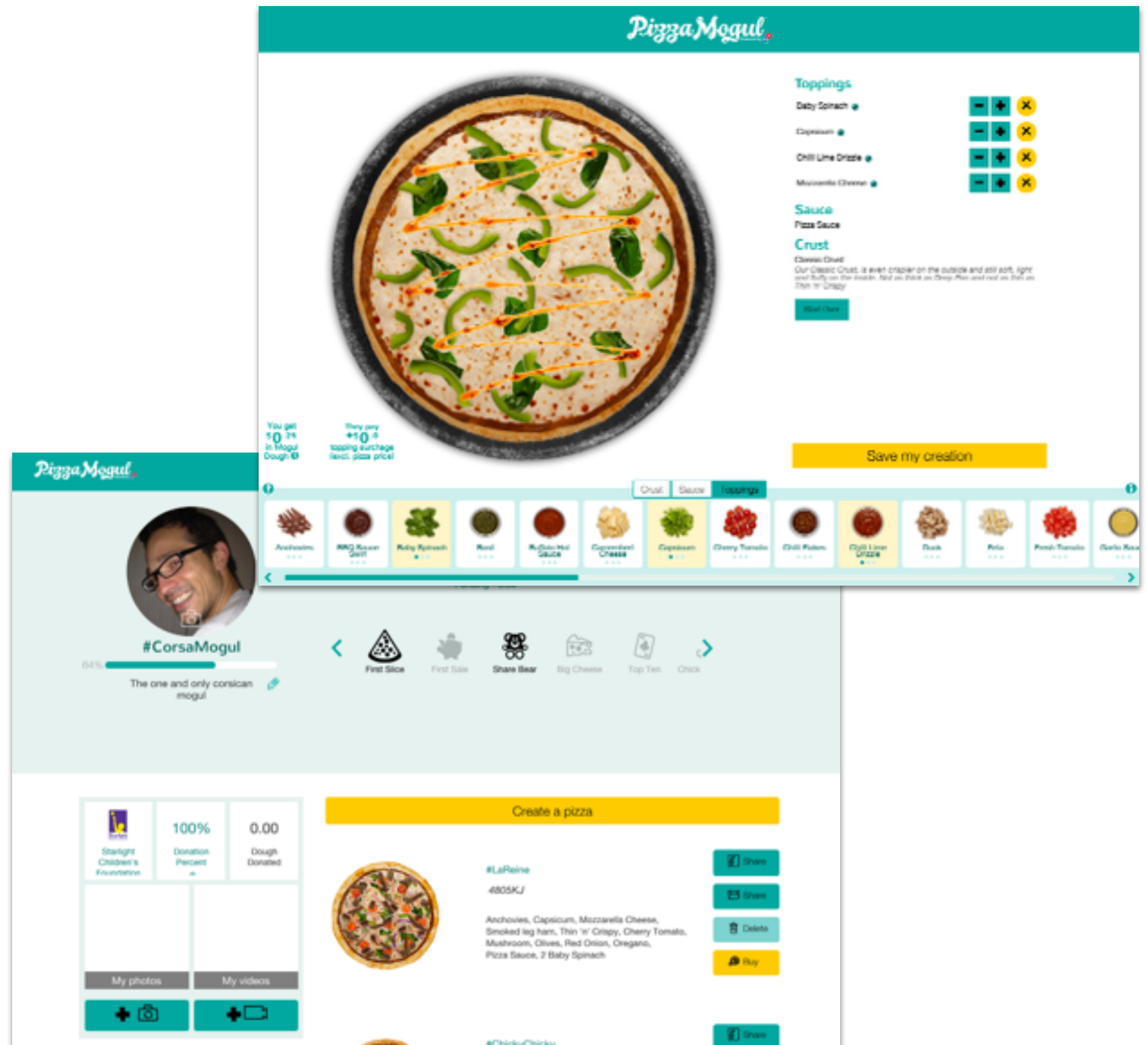
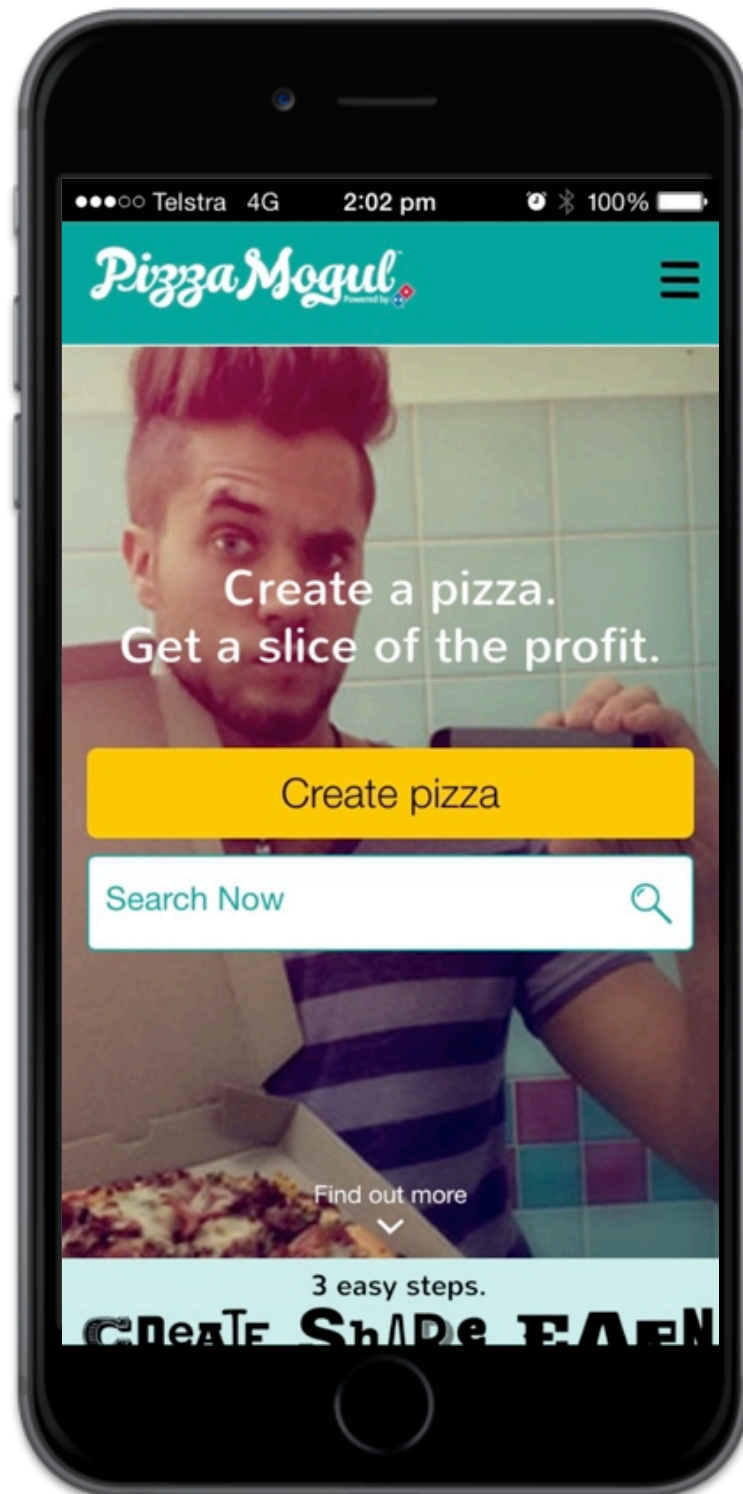


FROM THE TRENCHES

PIZZA MOGUL

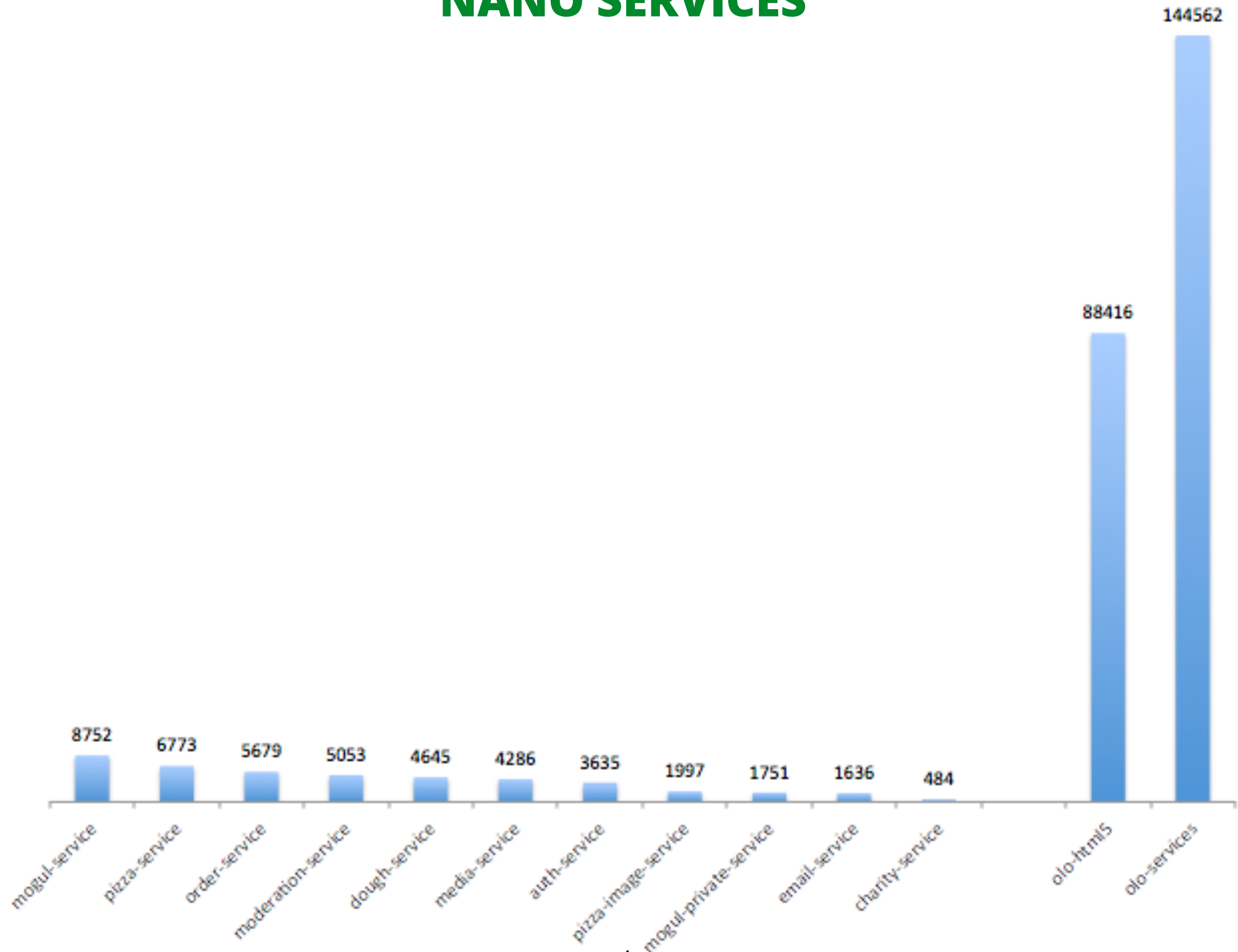
FROM THE TRENCHES

PIZZA MOGUL



FROM THE TRENCHES

NANO SERVICES



FROM THE TRENCHES

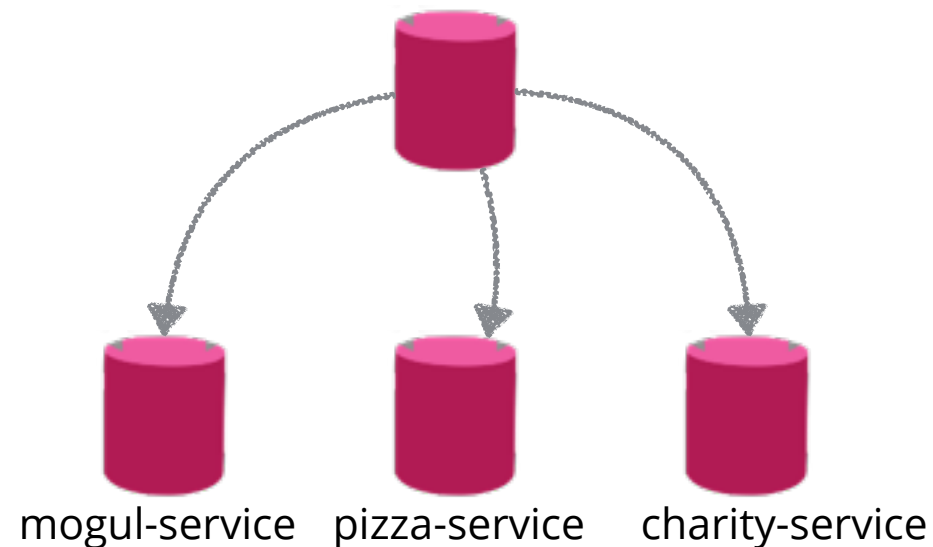
CROSS CUTTING CONCERNS

```
public void RefreshAllLeaderboards()  
{  
    [...]  
  
    WithConnection(  
        connection =>  
            connection.Execute(DbTable + "UpdateAllLeaderboards", CommandType.StoredProcedure));  
  
    [...]  
}
```

leaderboard-service



UpdateAllLeaboards



FROM THE TRENCHES

REFLECTIONS



microservices?



need big upfront design?

An aerial night photograph of a modern, multi-level bridge spanning a body of water. The bridge is illuminated with warm white lights, and its reflection is visible in the dark water. In the upper right corner, there are several colorful, curved light trails in blue, red, and yellow, suggesting long-exposure photography of moving lights or vehicles. The overall scene is dark, with the bridge lights providing the primary illumination.

DEFINE MICROSERVICES

**WHEN AGILE MEET DISTRIBUTED
SYSTEMS**

DEFINE MICROSERVICES

DISTRIBUTED AND WEB ARCHITECTURES

ANSI C

C++

Java

XML

JavaScript

1980s

1990s

2000s

2010s

TCP/IP

Client-Sever
RPC

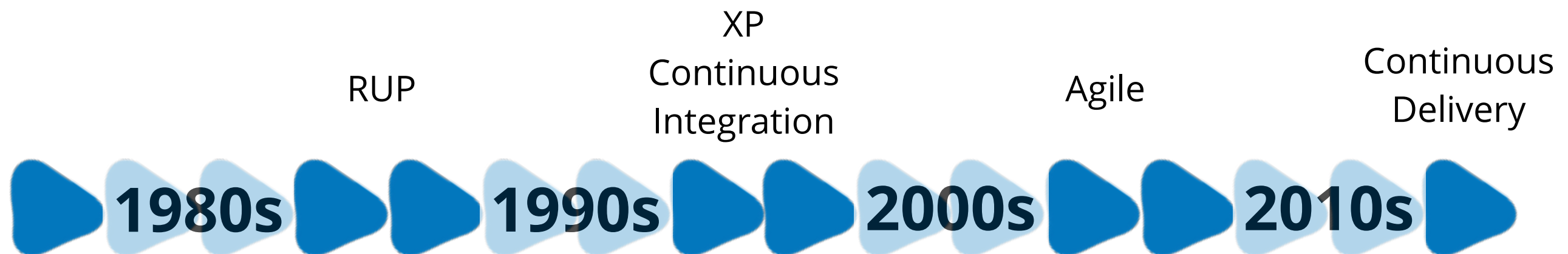
CORBA
RMI

SOAP
SOA

JSON
REST

DEFINE MICROSERVICES

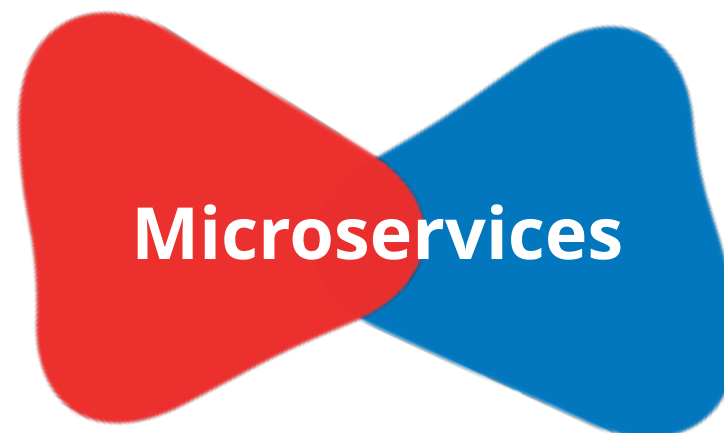
SOFTWARE DEVELOPMENT METHODOLOGIES



DEFINE MICROSERVICES

XXXXXXXXXXXXXXXXXX

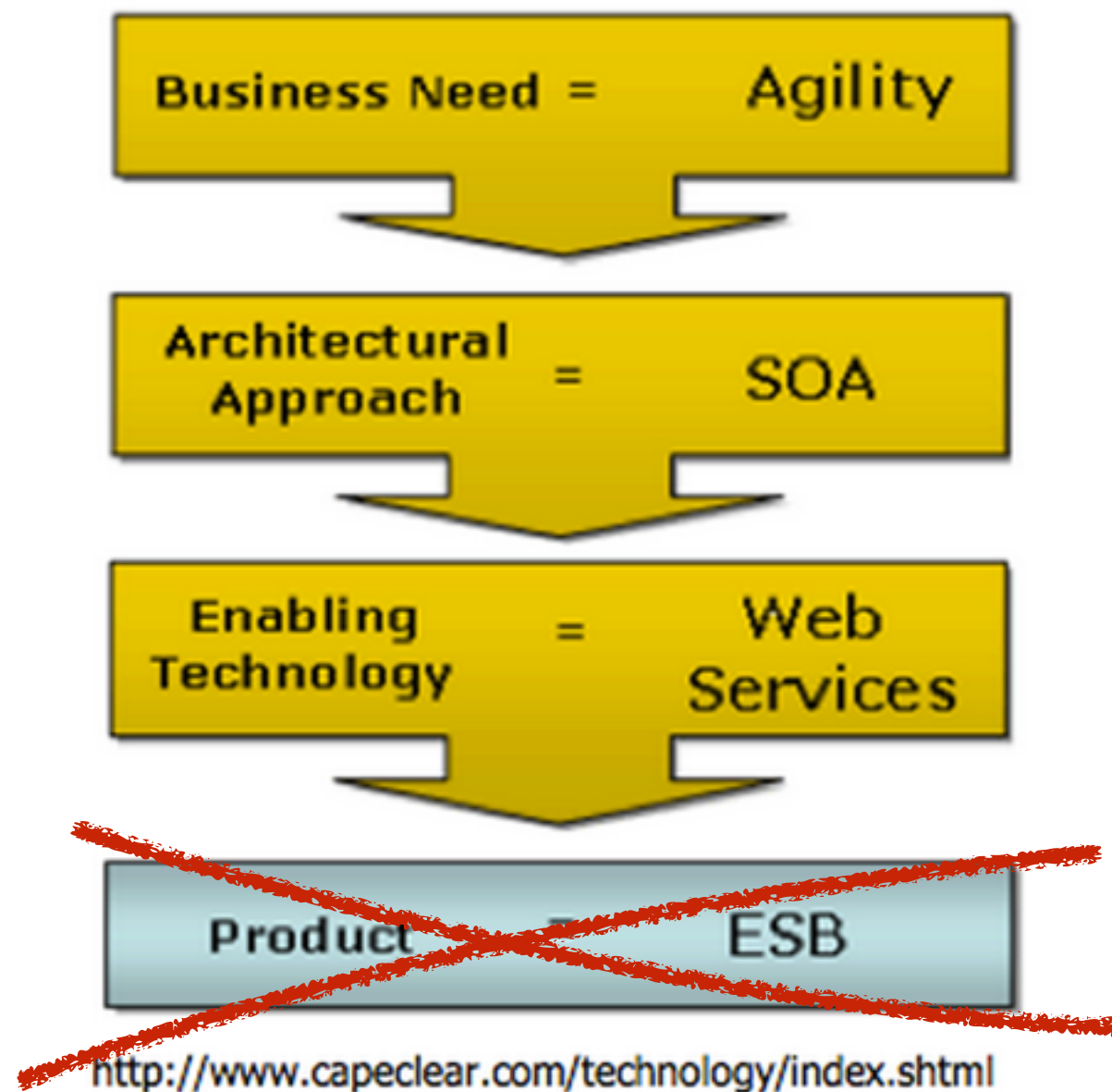
Distributed &
Web
Architecture



Agile &
Continuous
Delivery

DEFINE MICROSERVICES

AN ALTERNATIVE TO TRADITIONAL SOA



Microservices

DEFINE MICROSERVICES

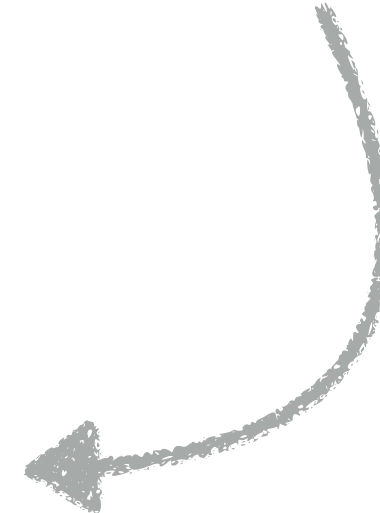
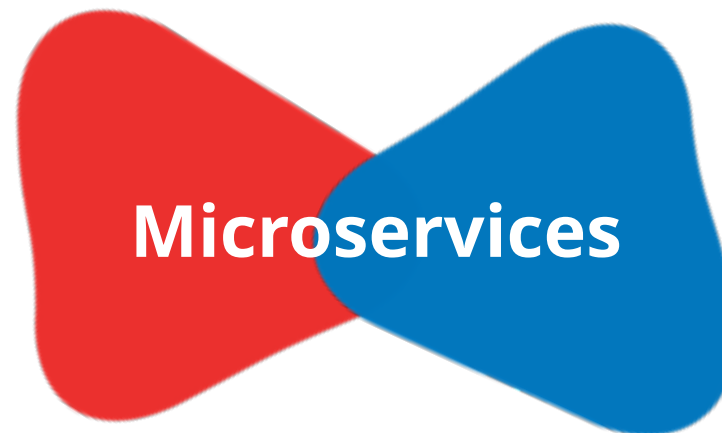
REFLECTIONS



microservices?



need big upfront design?





MICROSERVICES RUSH

WHAT MAKE THEM SO POPULAR?

MICROSERVICES RUSH

INTEREST



“microservices” searches

MICROSERVICES RUSH

SCALE

NETFLIX

"In order to scale to the next order of magnitude, we're rethinking the fundamentals of our architecture"

"We created the viewing service to encapsulate the domain of viewing data collection, processing, and providing"

"These components would be easier to develop, test, debug, deploy, and operate if they were extracted into their own services."

Philip Fisher-Ogden, Director of Engineering

MICROSERVICES RUSH

BREAK THE MONOLITH



Monolithic Rails —> JS / Rails / Scala —> microservices



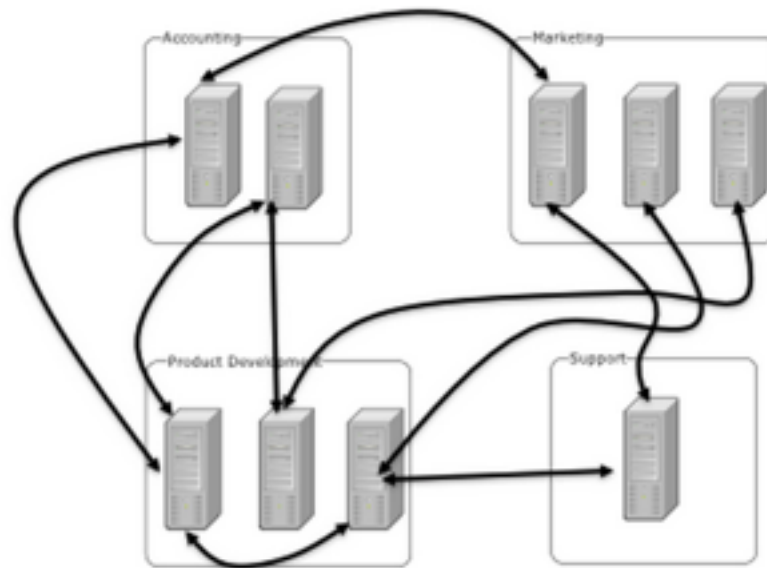
Monolithic Perl —> Monolithic C++ —> Java —> microservices



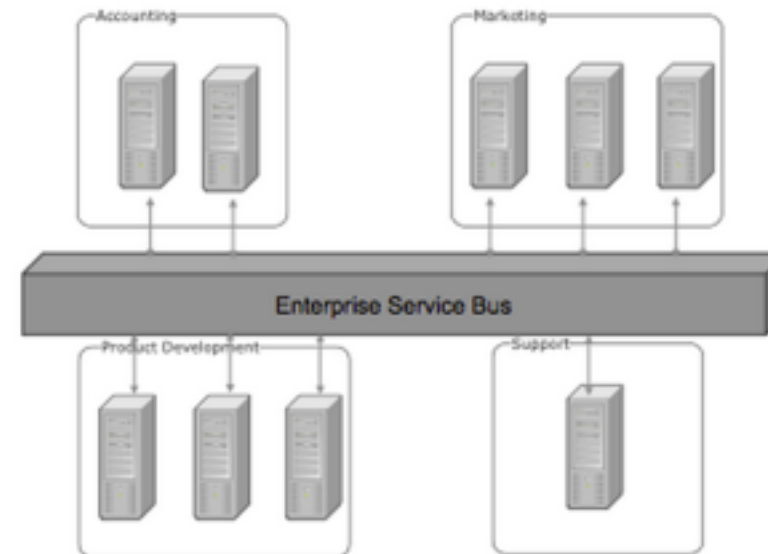
Monolithic C++ —> Perl / C++ —> Java / Scala -> microservices

MICROSERVICES RUSH

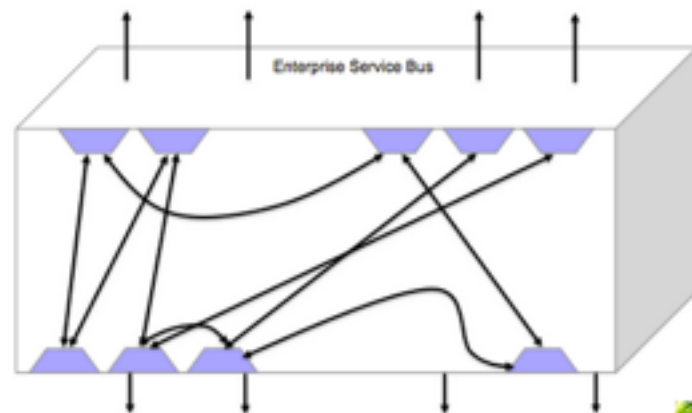
ENDPOINTS AS FIRST CLASS CITIZEN



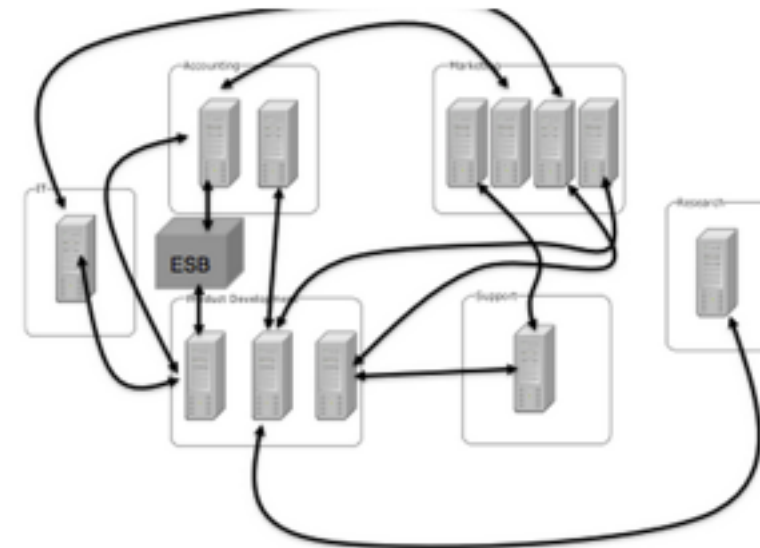
THE PROBLEM



THE SOLUTION?



THE PROBLEM (AGAIN)



THE FUTURE

MICROSERVICES RUSH

BUSINESS VALUE FIRST

Where is Strategic Value?



MICROSERVICES RUSH

A NEW ORGANISATION

"Any organization that designs a system (defined more broadly here than just information systems) will inevitably produce a design whose structure is a copy of the organization's communication structure."

Melvin Conway, *How Do Committees Invent* - 1968

MICROSERVICES RUSH

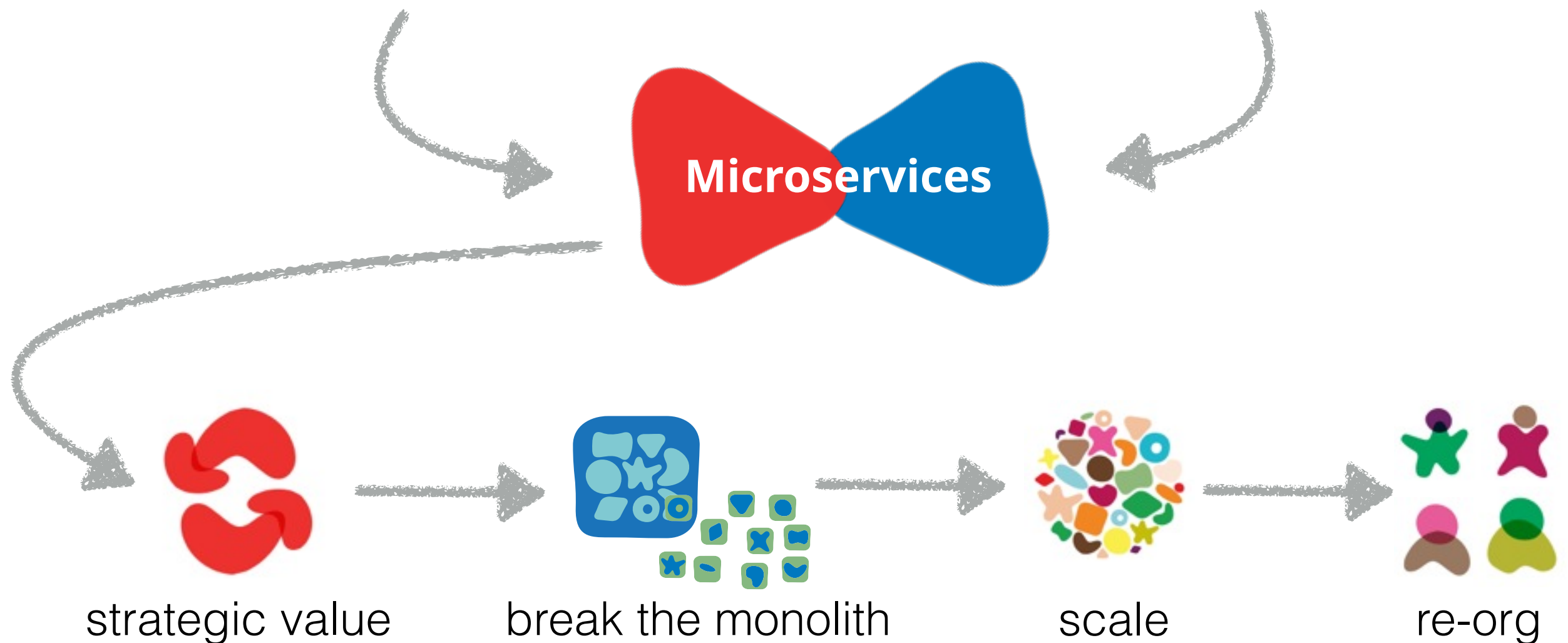
REFLECTIONS



microservices?



need big upfront design?



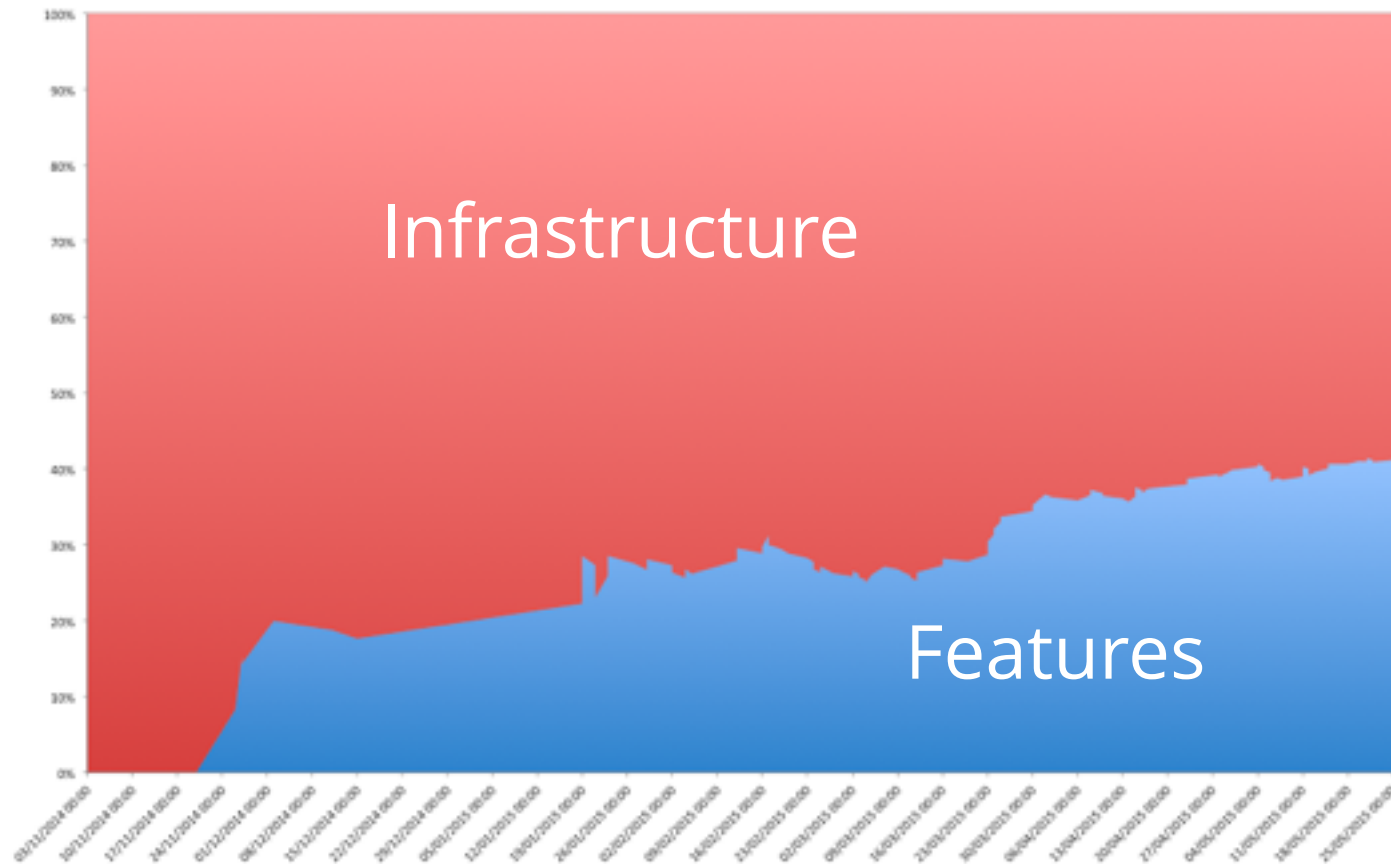


DELIVERING MICROSERVICES

IN AN AGILE WAY

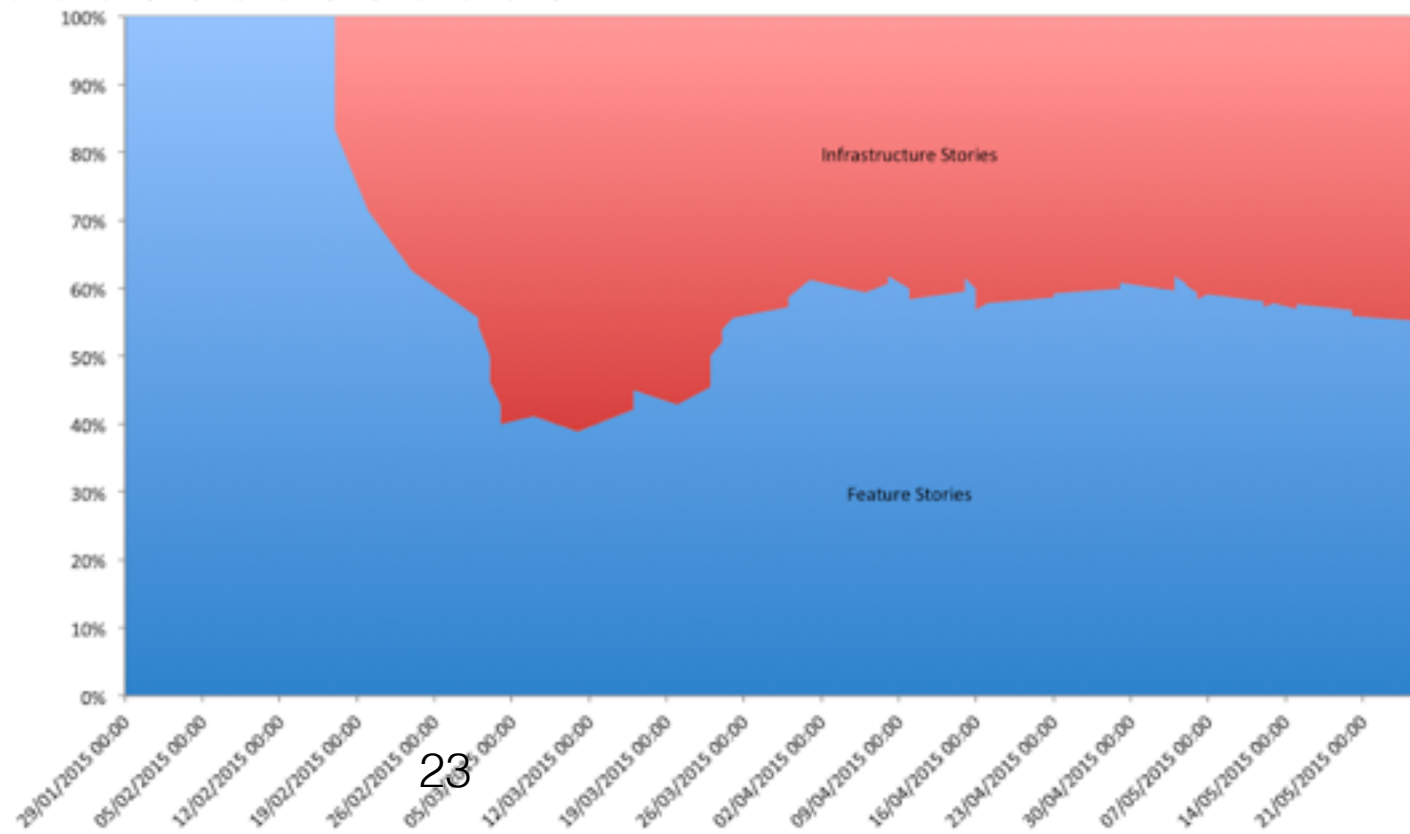
DELIVERING MICROSERVICES

THE PROBLEM



WatchlistService

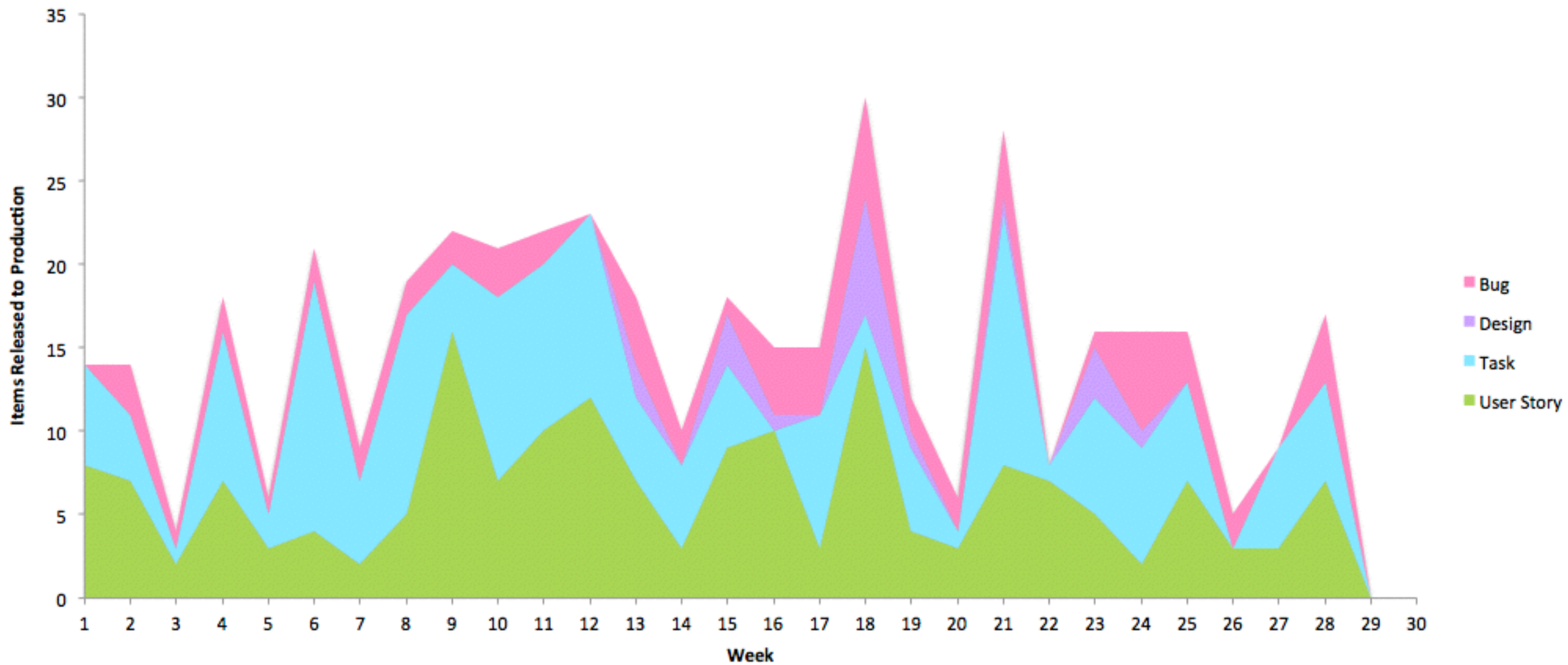
PriceEstimationService



DELIVERING MICROSERVICES

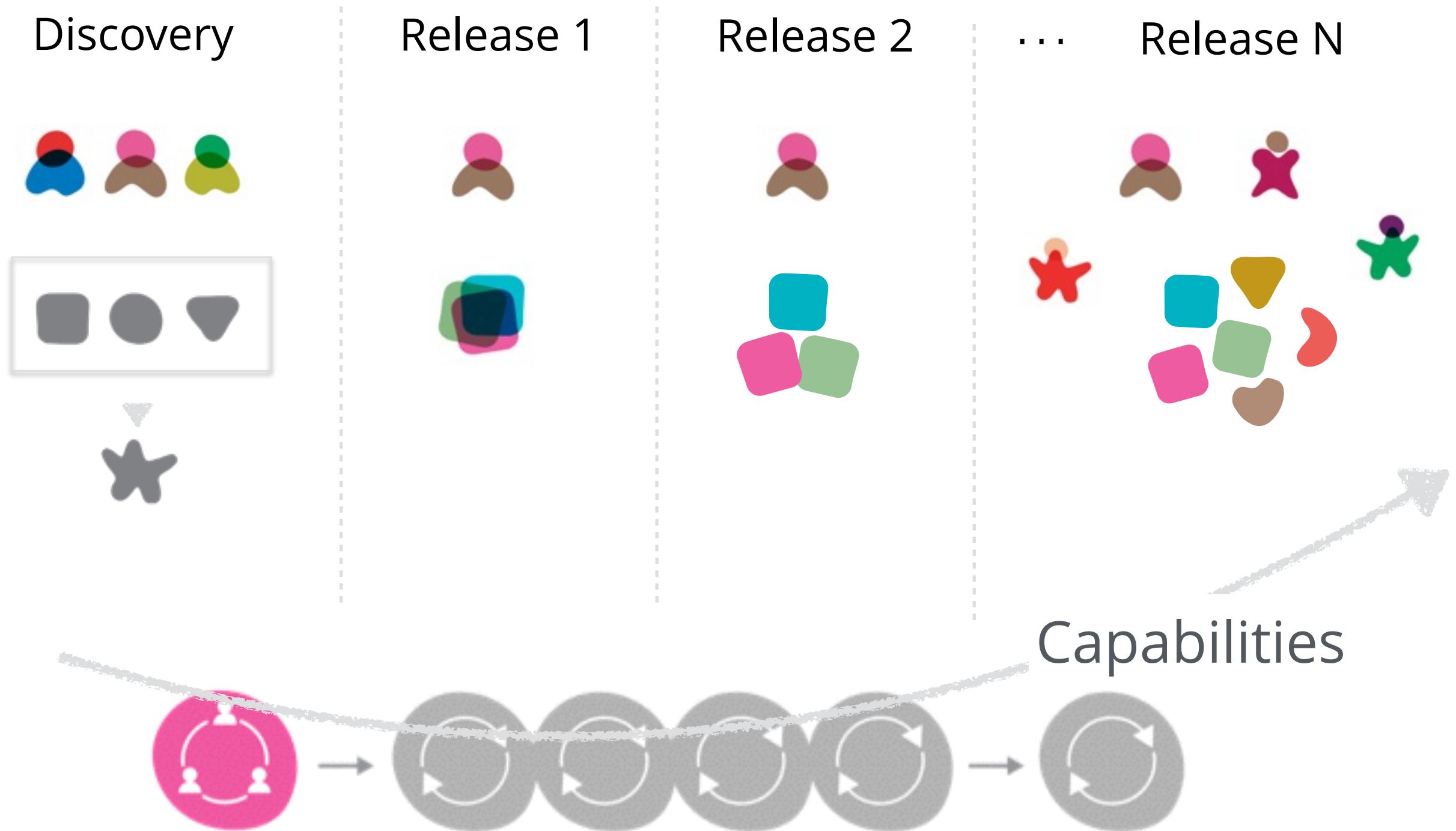
THE GOAL

Green Team - Items Released to Production - 2014



DELIVERING MICROSERVICES

JOURNEY



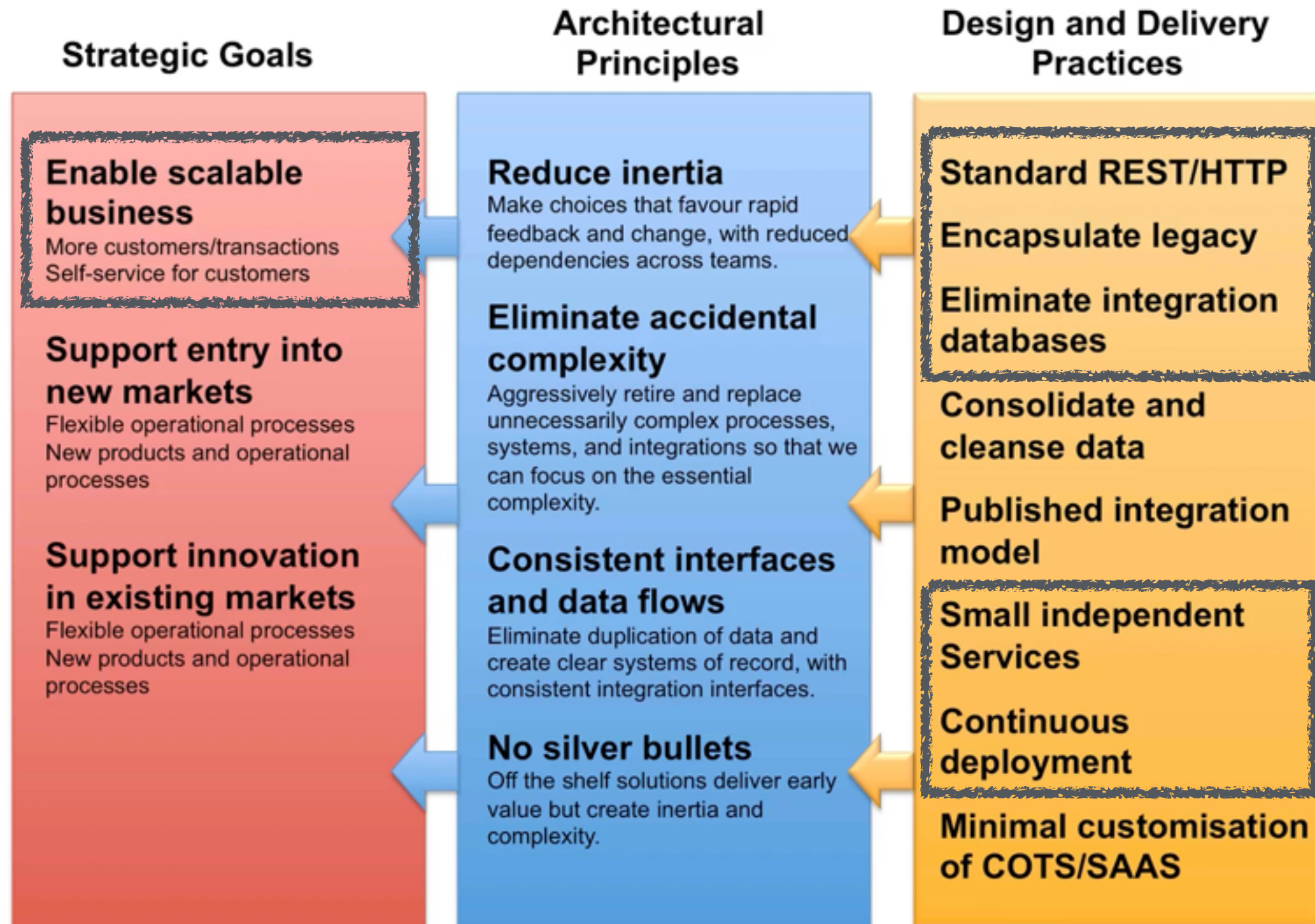
DELIVERING MICROSERVICES

DISCOVERY - WHY MICROSERVICES



DELIVERING MICROSERVICES

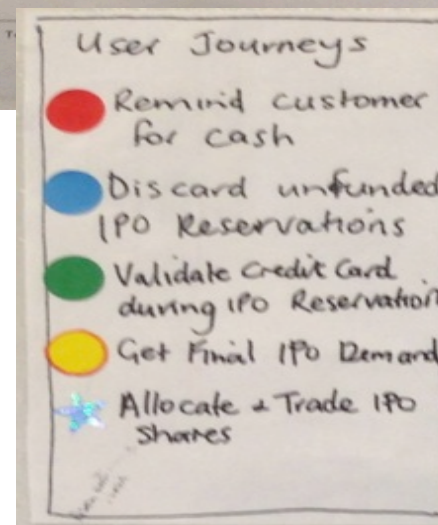
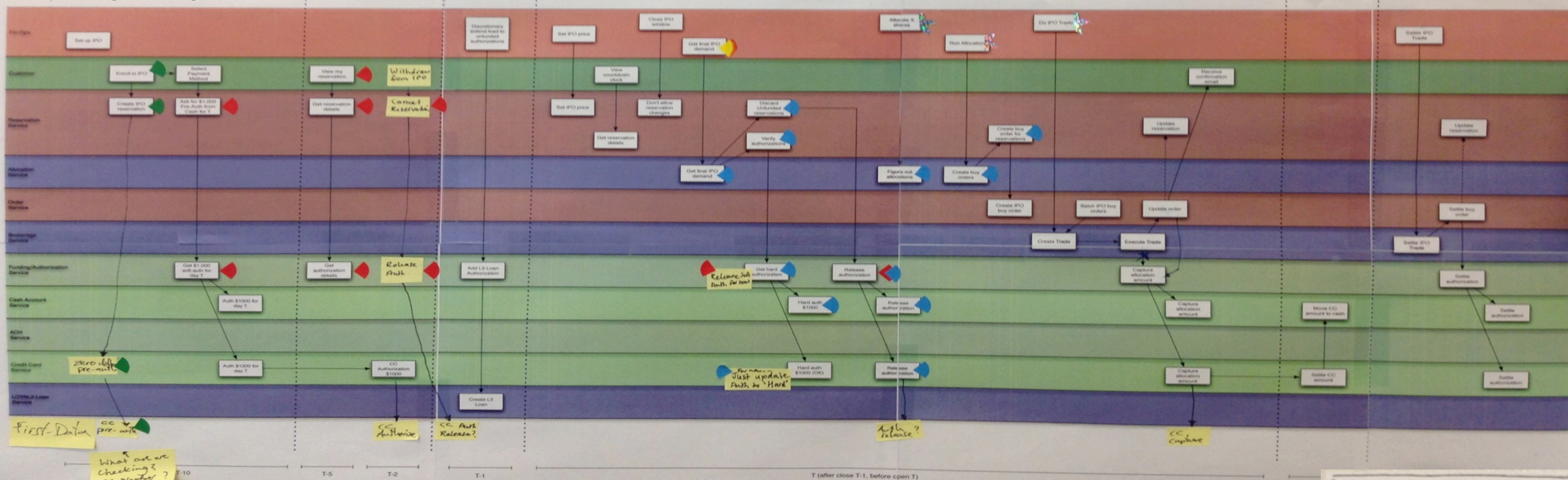
BUSINESS & ARCHITECTURE CONTRACT



DELIVERING MICROSERVICES

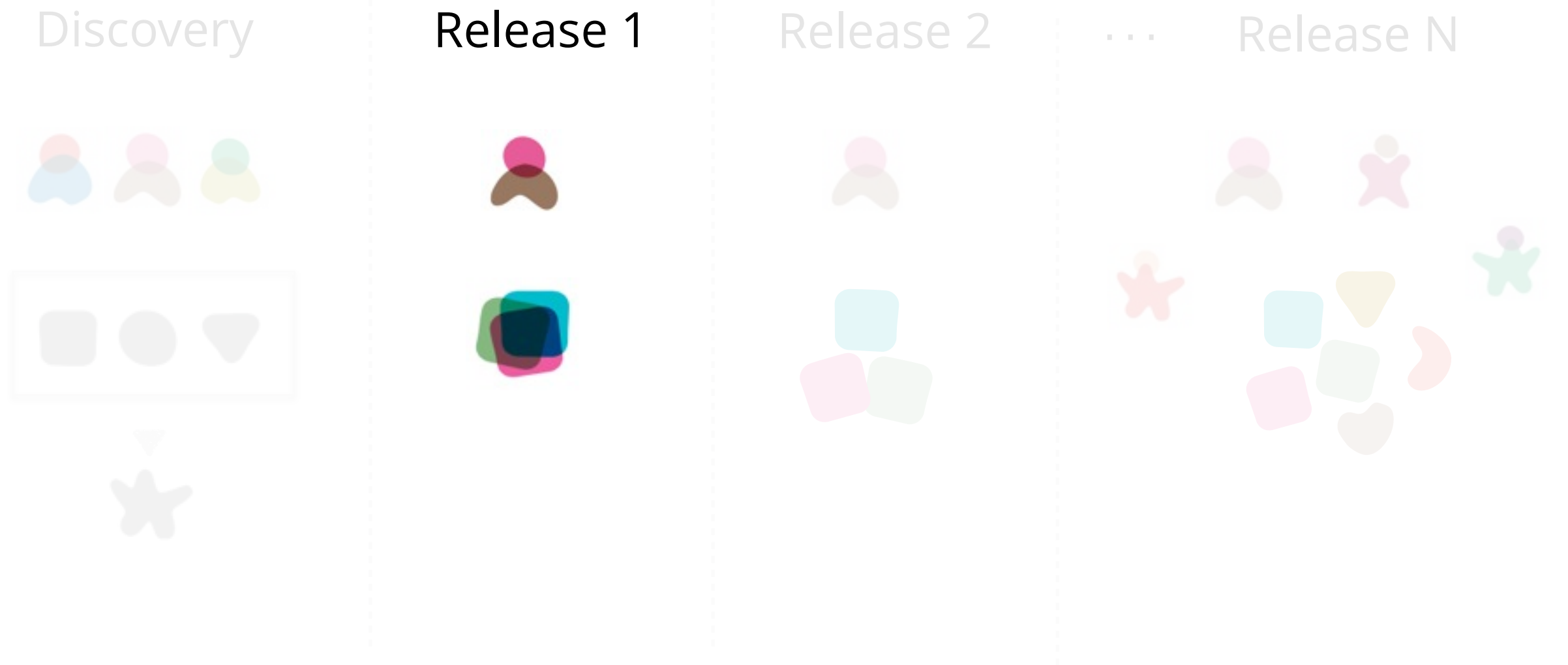
USER JOURNEYS ACROSS SERVICES

Decouple funding from trading



DELIVERING MICROSERVICES

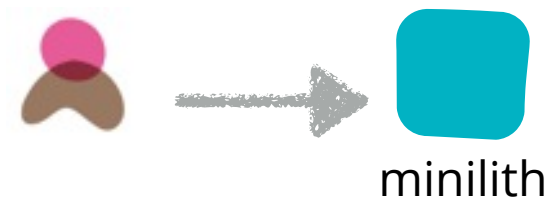
RELEASE 1 - BUILD A MINILITH



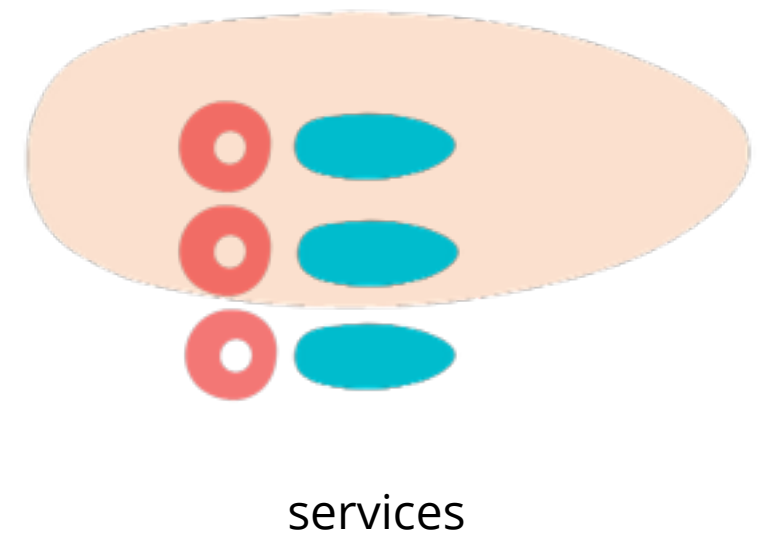
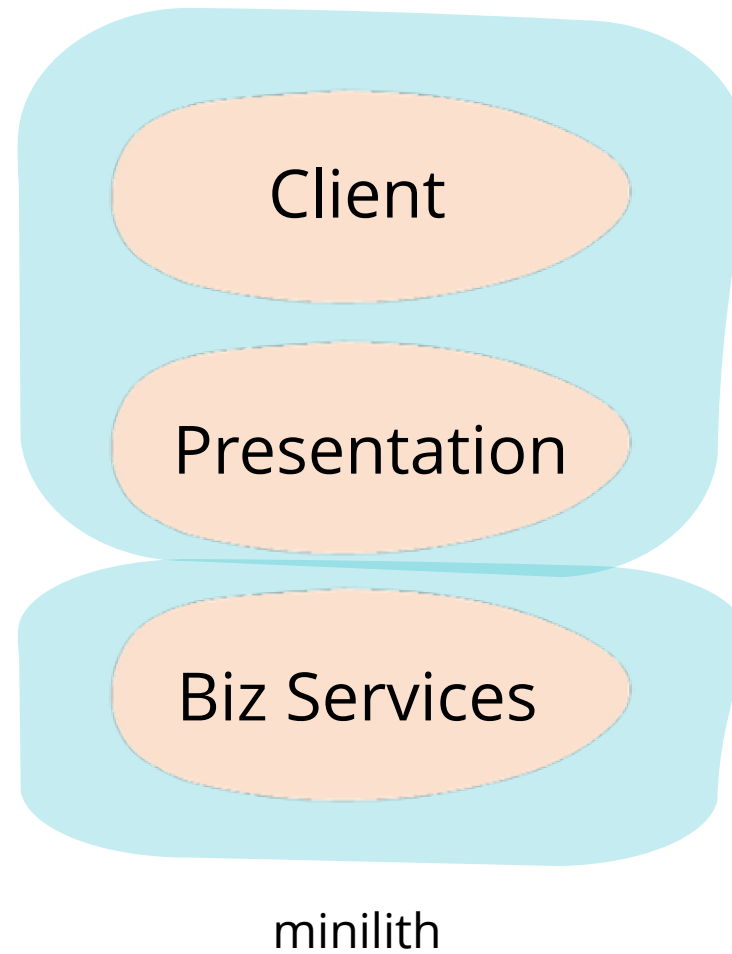
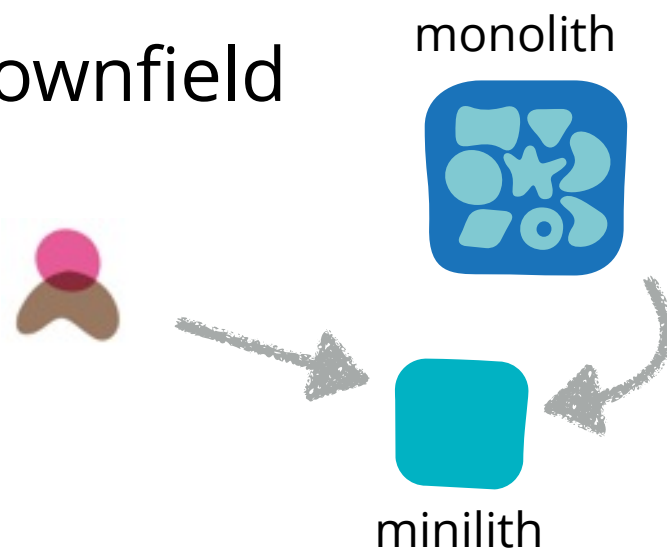
DELIVERING MICROSERVICES

RELEASE 1 - ARCHITECTURE

greenfield

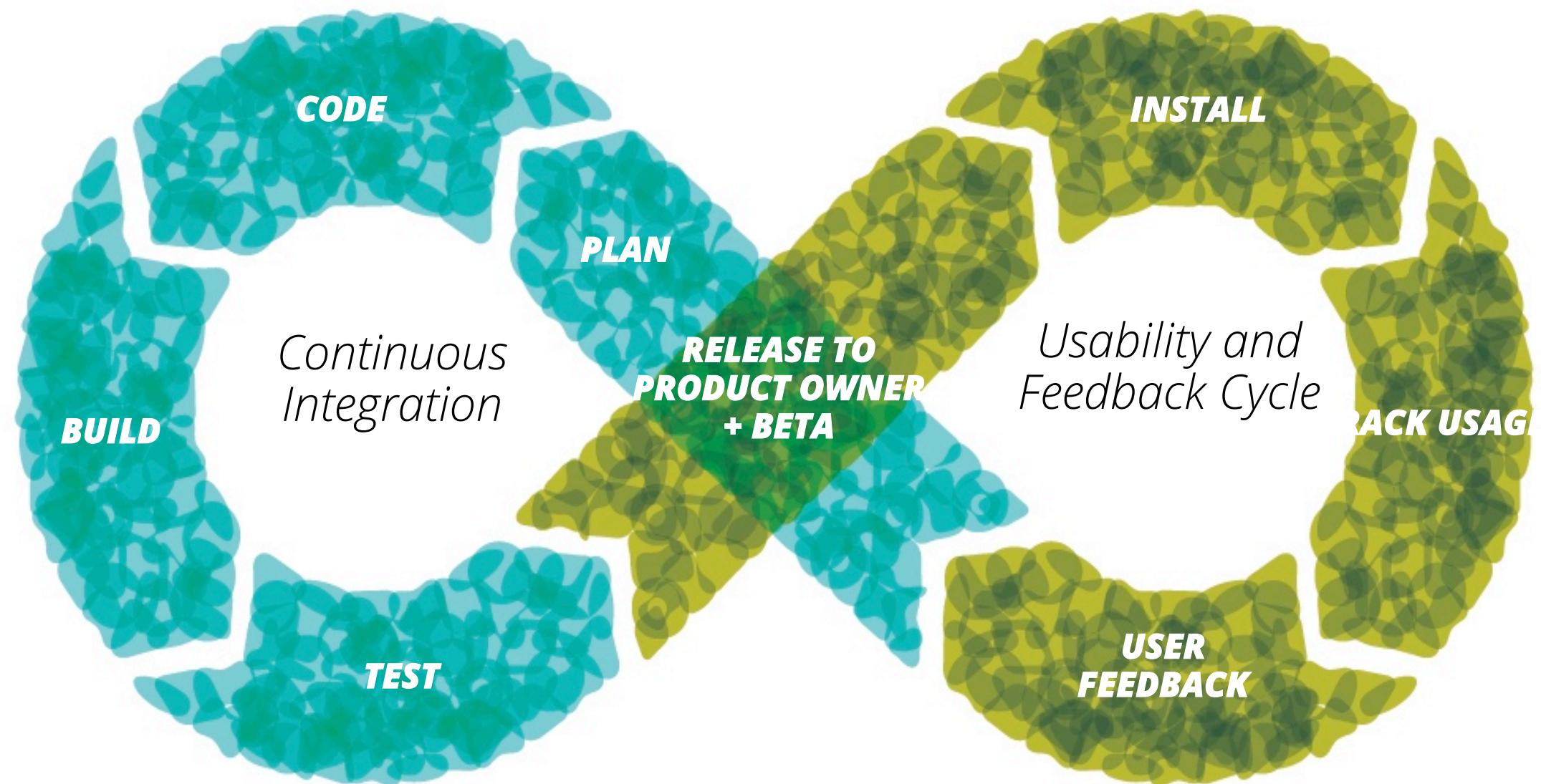


brownfield



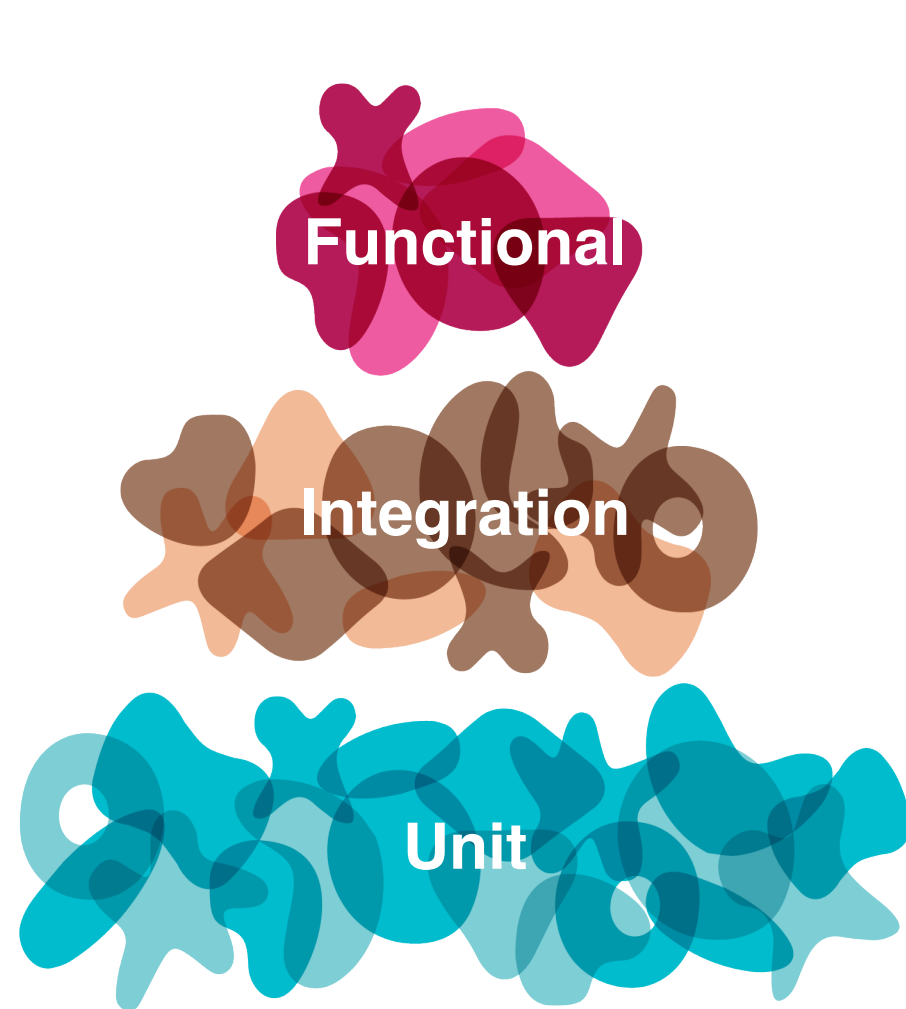
DELIVERING MICROSERVICES

RELEASE 1 - CONTINUOUS DELIVERY

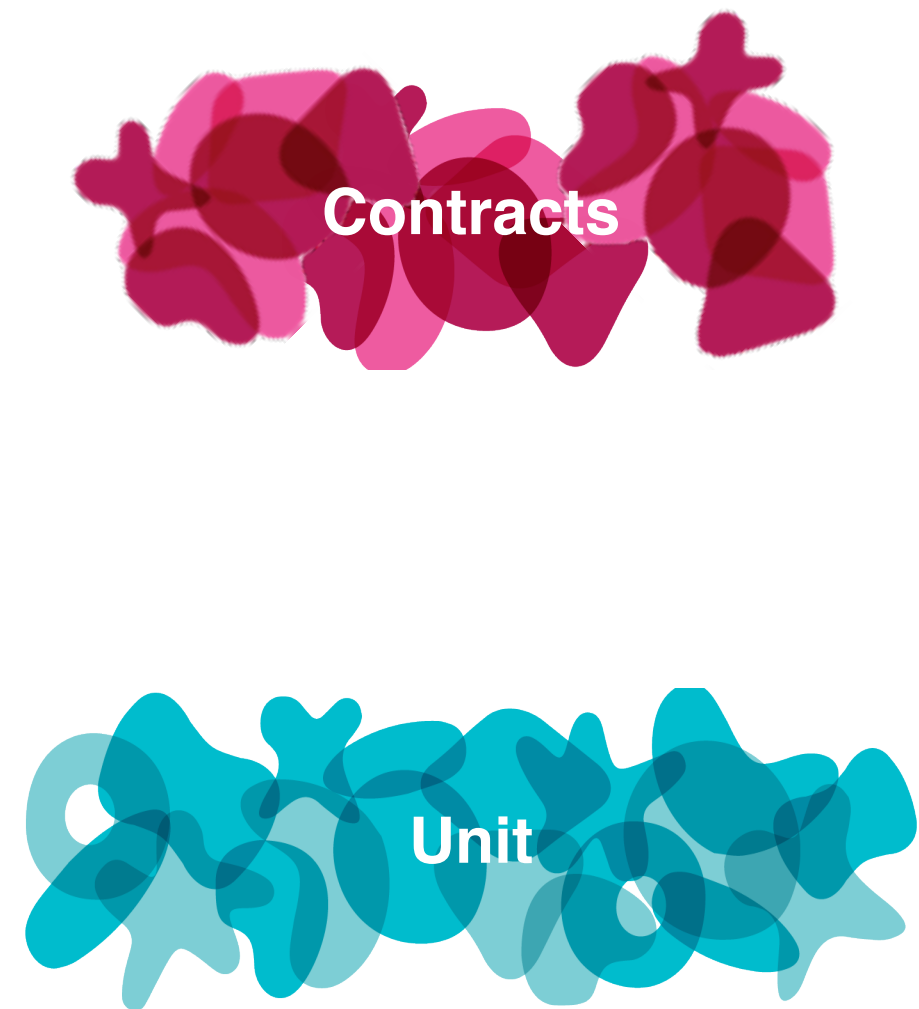


DELIVERING MICROSERVICES

RELEASE 1 - CONTRACT TESTING



Web Application



Web Services

DELIVERING MICROSERVICES

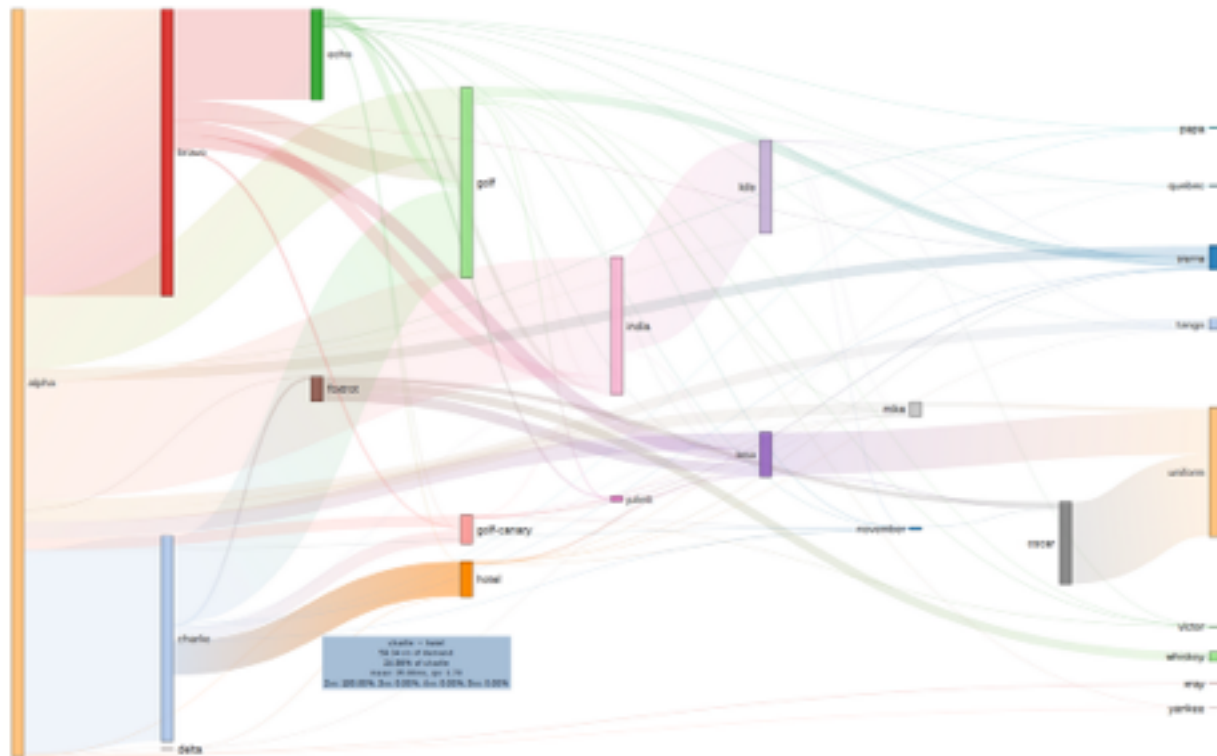
RELEASE 1 - CONTRACT TEST EXAMPLE

```
describe('with an email created', function() {

  beforeEach(function(done) {
    var requestOptions = {
      url: config.serviceUrl + '/badge',
      method: 'POST',
      json: {
        'mogulid': '*****',
        'badgecode': 'FirstSlice',
        'name': 'First Slice',
        'description': 'Awarded when a Mogul creates their first pizza',
        'imageurl': '/images/badges/firstslice.png',
        'bonusamount': '0'
      }
    };
    request.get(requestOptions, function(err, response, body) {
      postResponse = response;
      postBody = body;
      getEmails(done);
    });
  });

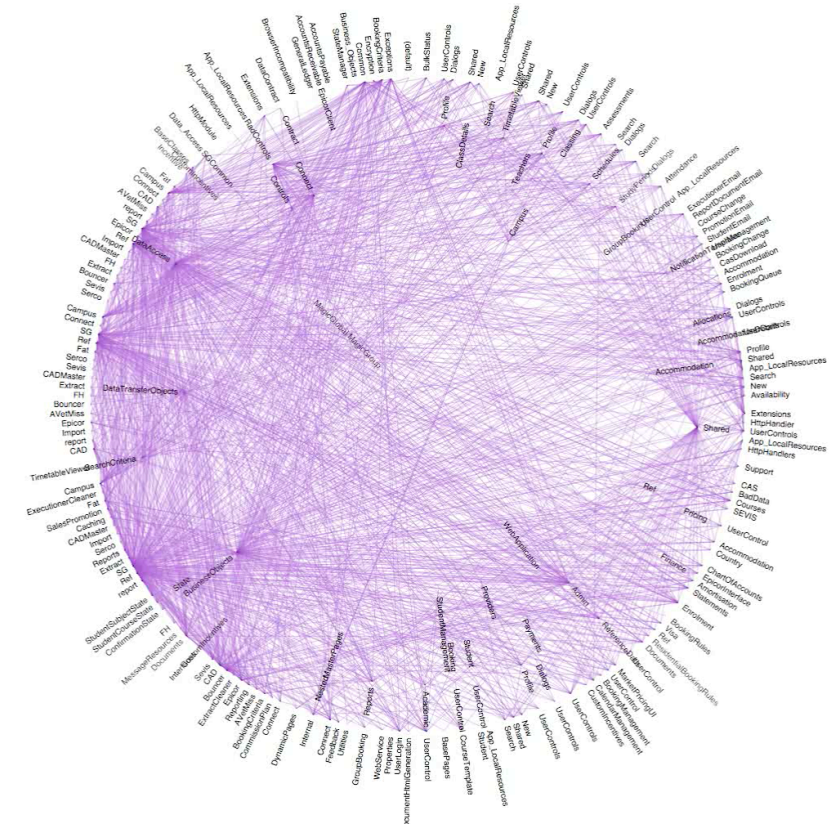
  it('returns the email', function() {
    expect(postResponse.statusCode).toBe(201);
    var xml = new dom().parseFromString(getBody);
    expect(xpath.select('count(//*[local-name()=\'MogulEmail\'] )', xml)).toBe(1);
    expect(xpath.select('count(//*[local-name()=\'Field\'] )', xml)).toBe(9);
    expect(xpath.select('//*[local-name()=\'MogulEmail\']', xml)[0].namespaceURI)
      .toBe('http://PizzaMogul.Integration.Communication.Schemas.MogulEmail');
  });
});
```

RELEASE 1 - DEPENDENCY GRAPHS WITH D3JS



D3 Sankey Diagram

<http://techblog.netflix.com/2015/02/a-microscope-on-microservices.html>

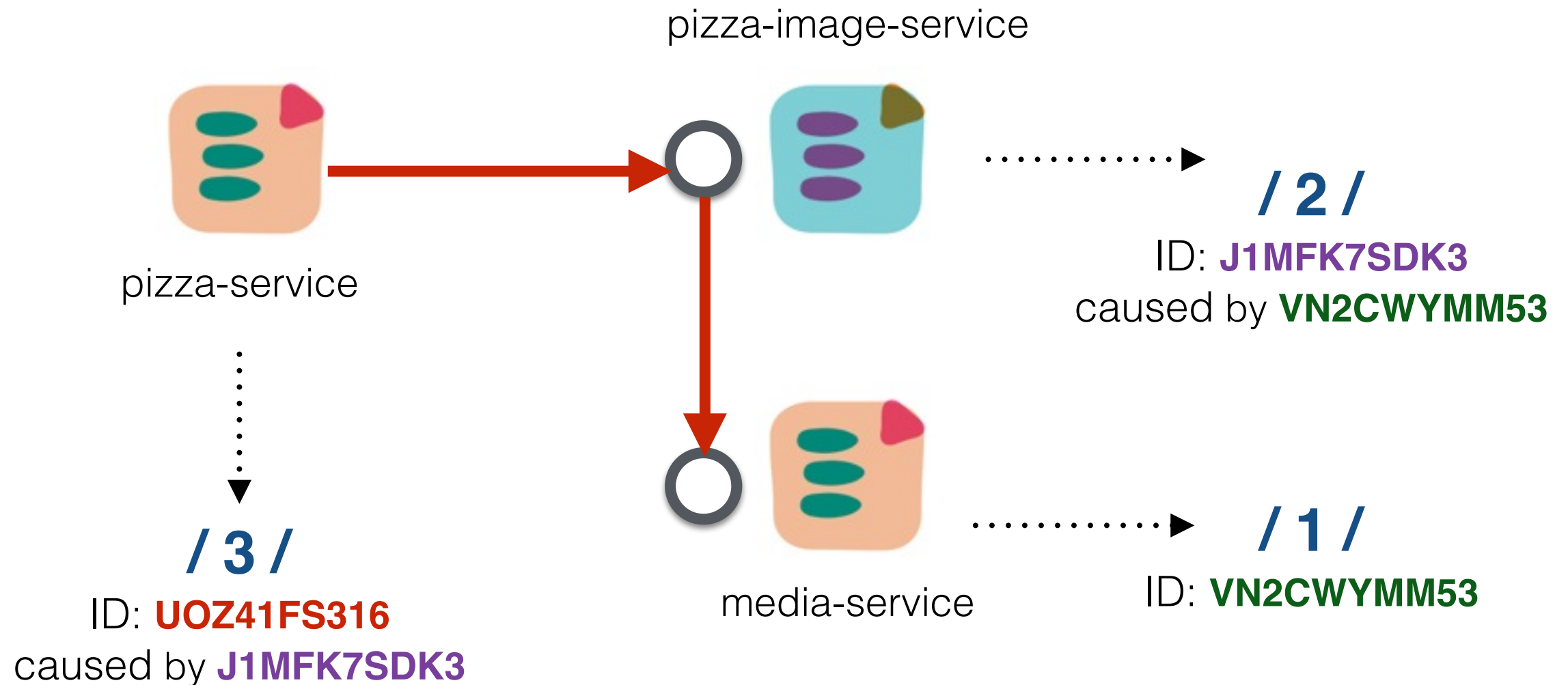


D3 Hierarchic Edge Bundling

<https://www.youtube.com/watch?v=hsoovFbpAoE>

DELIVERING MICROSERVICES

RELEASE 1 - TRANSACTIONS LOGGING



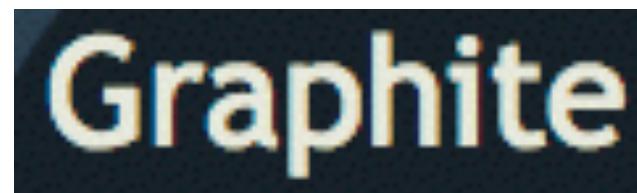
DELIVERING MICROSERVICES

RELEASE 1 - DIFFERENT LEVELS OF MONITORING

Log



Runtime



Infrastructure



DELIVERING MICROSERVICES

RELEASE 1 - BUILD CAPABILITIES

- **Continuously Deliver**
CI, automated builds, tests, and deployments, release management
- **Re-think the pyramid of tests**
Favor contracts testing to UI functional testing
- **Control dependencies**
Modules integration via HTTP, dependency graph, transaction logging
- **Release to production**
Learn about your prod environment, consider containers and/or in-process lightweight web servers, load balancers, caching, etc...
- **Monitor & Analyse**
Have system monitoring and data analytics in place.

DELIVERING MICROSERVICES

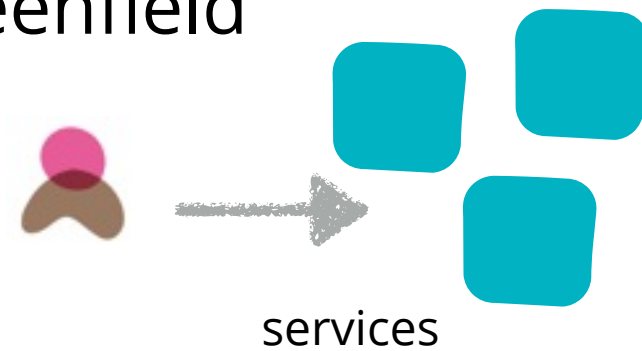
RELEASE 2 - BREAK THE MINILITH



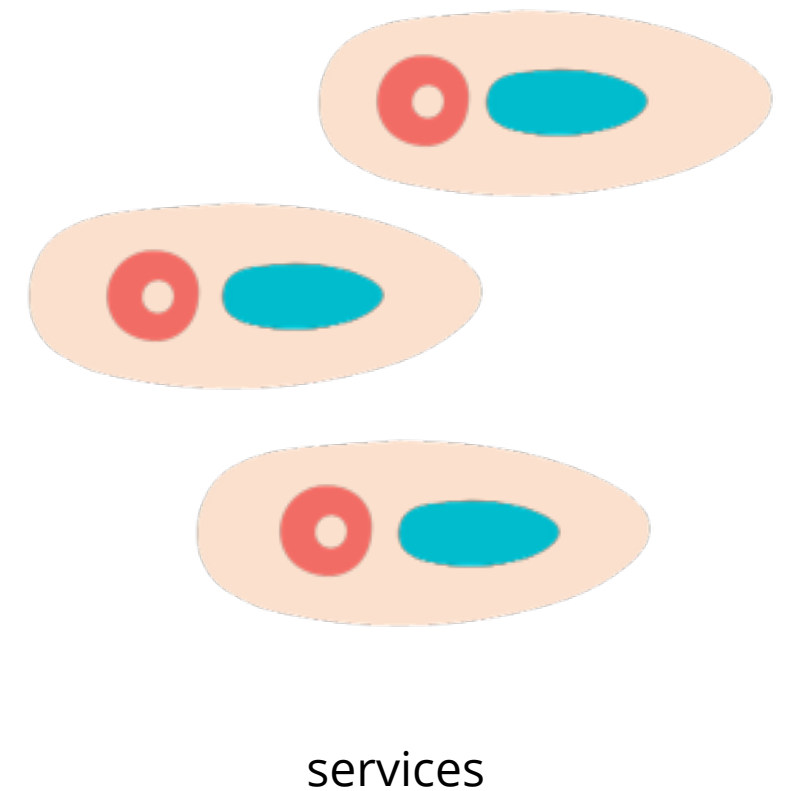
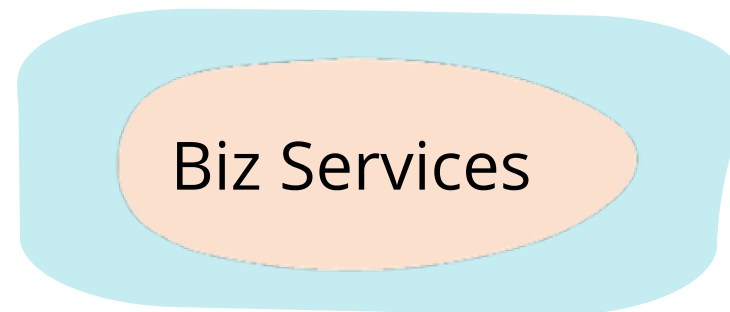
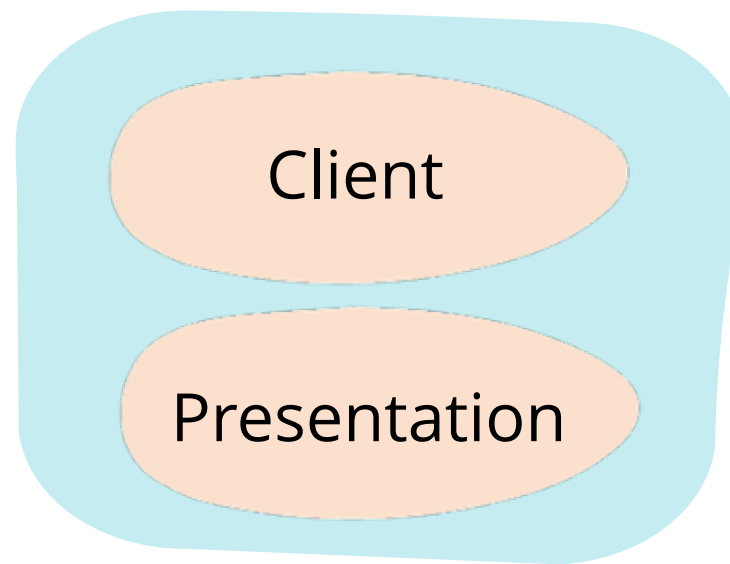
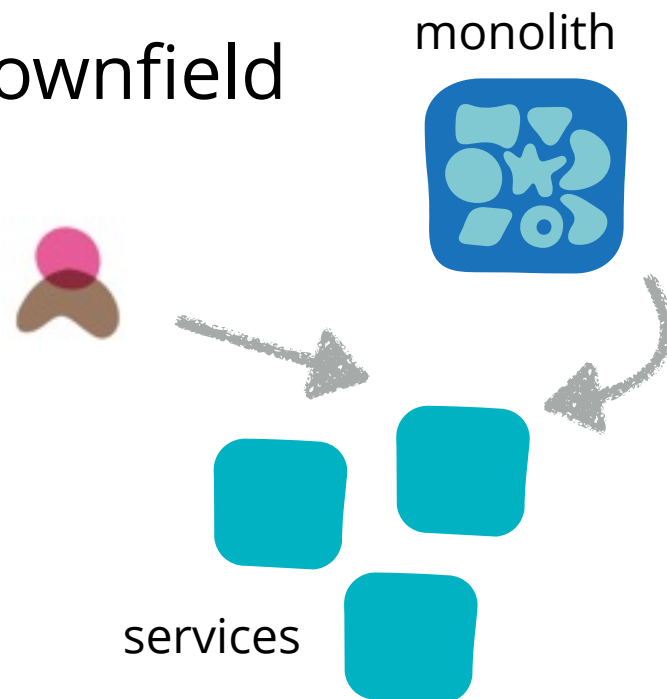
DELIVERING MICROSERVICES

RELEASE 2 - ARCHITECTURE

greenfield

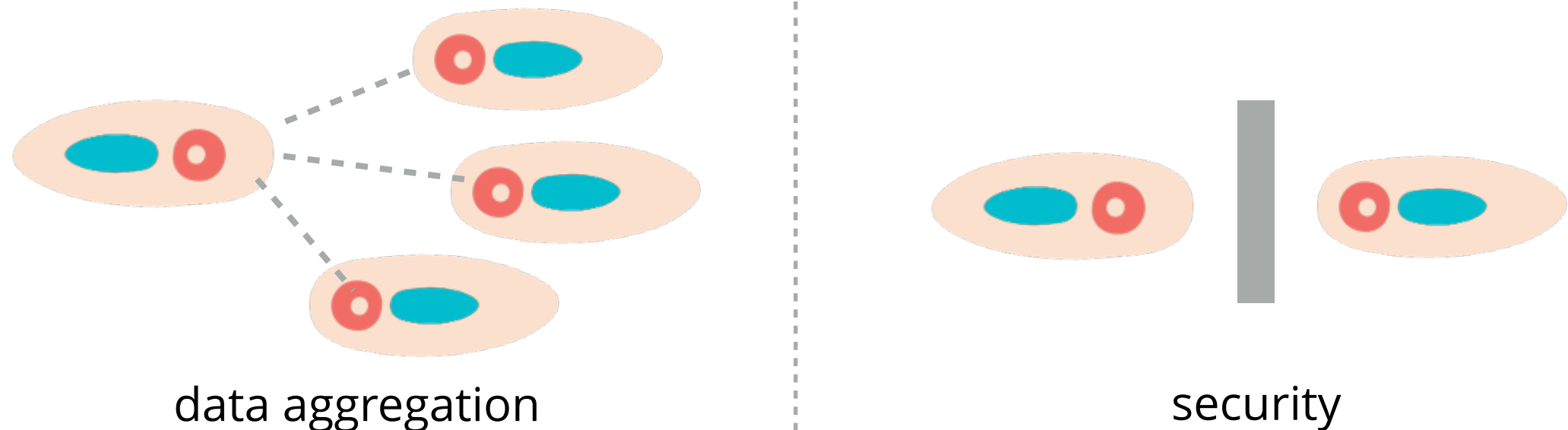
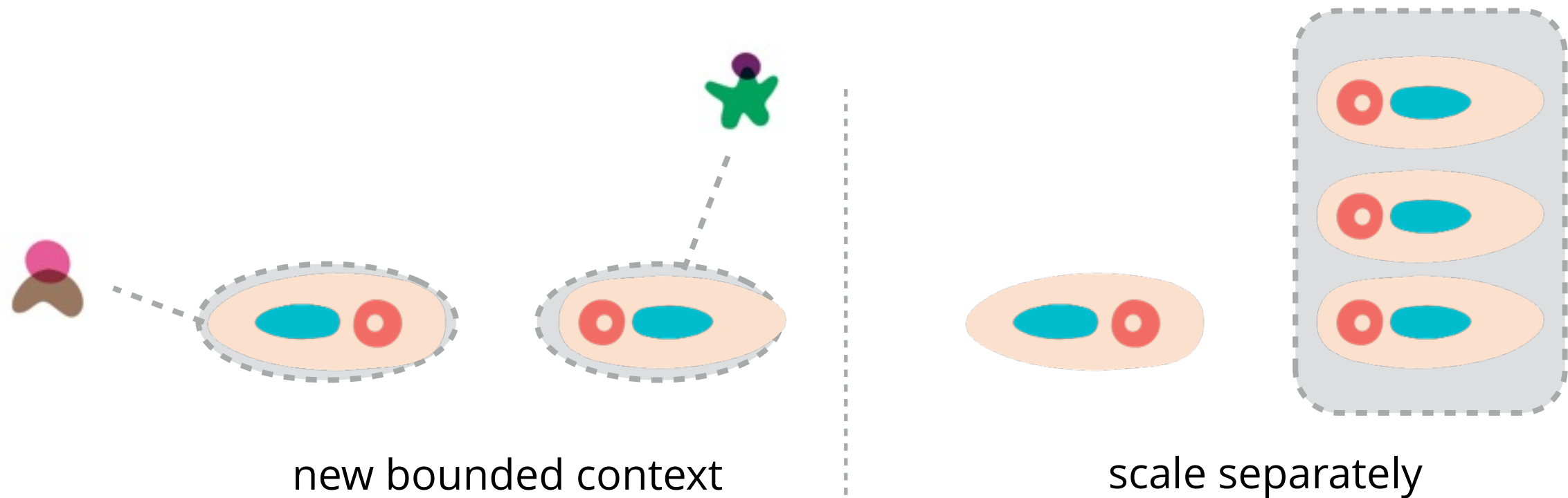


brownfield



DELIVERING MICROSERVICES

RELEASE 2 - NEW SERVICE USE CASES



DELIVERING MICROSERVICES

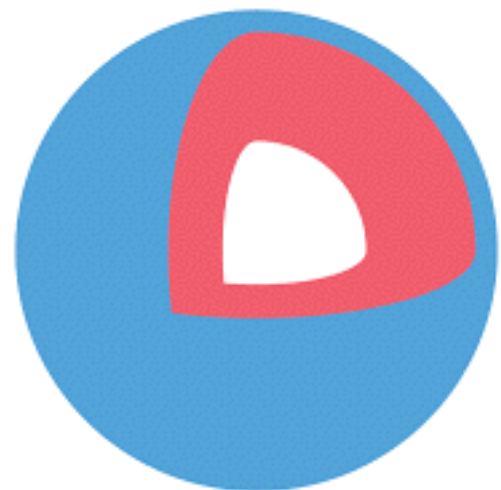
RELEASE 2 - ISOLATION DATA

"No direct database access is allowed from outside the service, and there's no data sharing among the services"

Werner Vogels, CTO and Vice President of amazon.com

DELIVERING MICROSERVICES

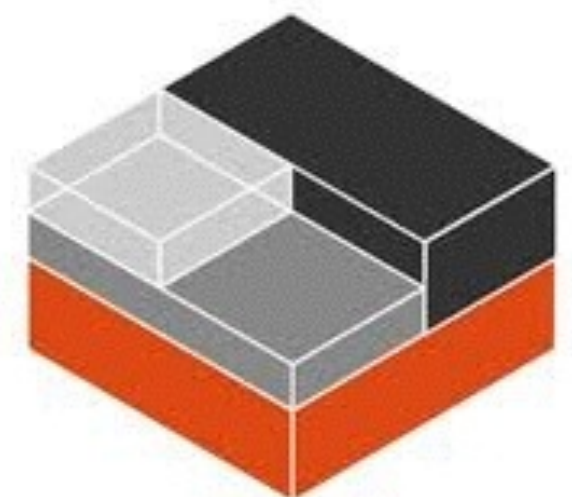
RELEASE 2 - ISOLATION CONTAINERS



Core OS



docker



LXD

DELIVERING MICROSERVICES

RELEASE 2 - ISOLATION INTEGRATION

Orchestration



Choreography



DELIVERING MICROSERVICES

RELEASE 2 - SERVICE TEMPLATES

FOLDERS

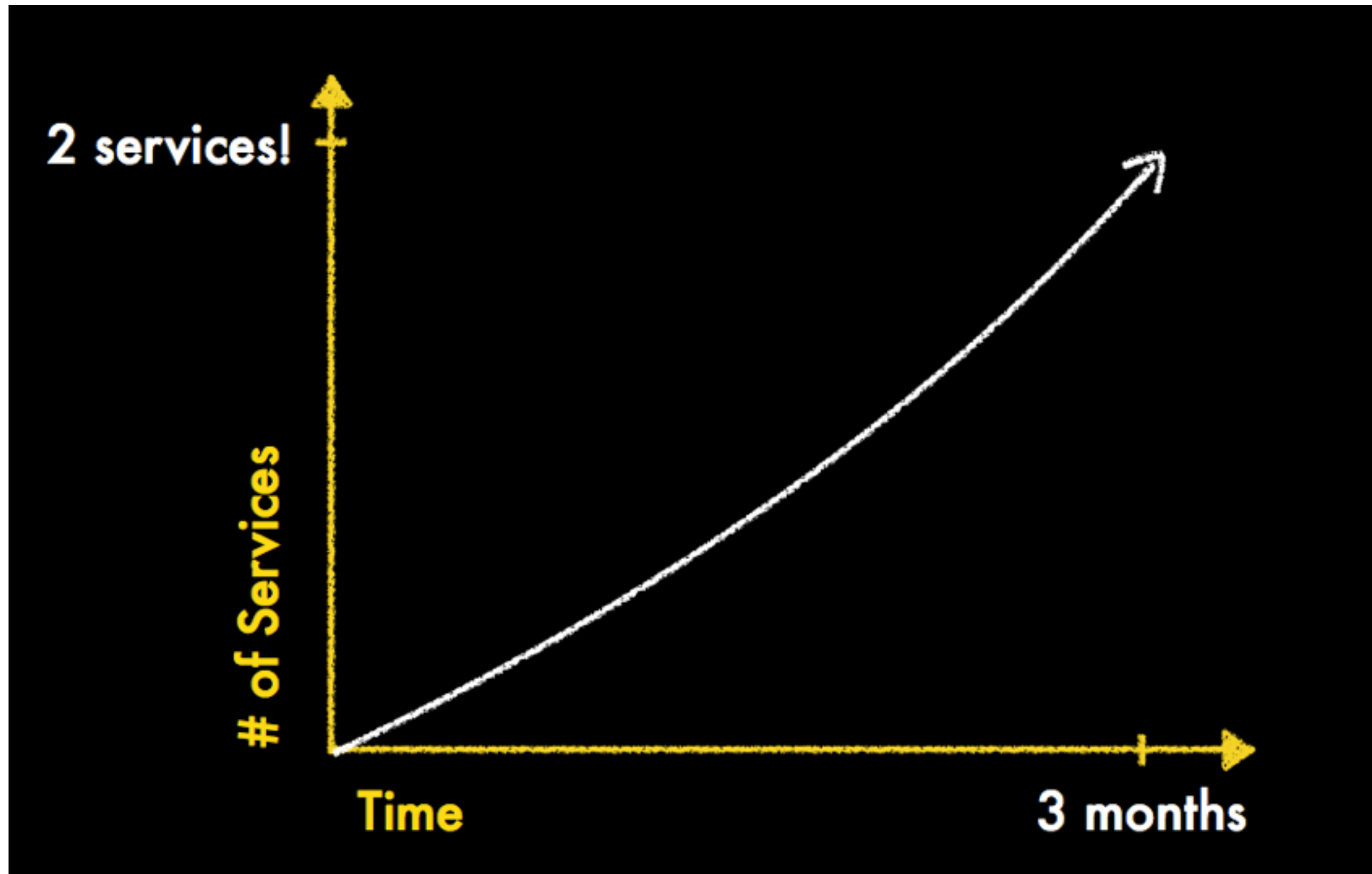
- ▼ Working
 - ▶ blah-service
 - ▶ blah-service-test
- README.md

README.md

```
1 service-template
2 =====
3
4 In your service, <BlahService> would be renamed by your actual service (e.g.
5 MyService).
6
7 #Repository Setup
8 ##Github Repository
9
10 1. Check out service-template repo (if you have not already done so)
11
12 1. Copy blah-service and blah-service-test directories into working directory
13
14 1. Rename each directory and their contents to whatever shiny new name you want
15
16     You will also need to do some fancy Visual Studio renaming
17
18     a. Update names in the .sln file in a text editor
19
20     b. Update names in the .csproj files in a text editor
21
22     c. Update the namespace. In Visual Studio in each project (Src and Test), go to
23     Project > ProjectName Properties and change Assembly Name and Default Namespace
24
25 1. Do a search for "blah" throughout the service and service-test directories and
26     replace with the name of your service if required. This includes Web.config and its
27     transformations.
28
29 1. If the service will be deployed to App servers in production (e.g. it is
30     internal only) then update 'Web.PreProd.config' and 'Web.Prod.config' to use the
31     correct 'MogulConfigFilePath' value.
32
33 1. In each directory, run
34
35     git init
36
37 1. create new repository (or repositories) on github (do this step for service and
38     test)
39
40 1. add the CI team to your service's repo (and your service test's repo) by going
41     to http://github.com/orgs/yemtu/teams/CI and click "+ Add Repository" (do this step
42     for service and test)
43
44 1. add the Development team to these same repos by going to https://github.
45     com/orgs/yemtu/teams/developers and click "+ Add Repository" (do this step for
46     service and test)
47
48 1. In each directory
49
50     git remote add origin https://github.com/yemtu/{repo-name}.git
51
52 1. Add and commit changes
53
54     git push -u origin master
55
56
57 ##TFS Repository
58
59 Note: requires git-tf
```


DELIVERING MICROSERVICES

RELEASE 2 - BUILD WHAT YOU CAN SUPPORT



DELIVERING MICROSERVICES

RELEASE 2 - UNDERSTAND SERVICES

- Know if and when to create a new service
new bounded context, aggregation, security, scale independently
- Define (your level of) isolation
code, compilation, runtime, persistence, integration
- Reduce barrier to entry
repository blueprint, pipeline template, scaffolding script
- Build what you can support
develop DevOps culture (e.g. infrastructure as code)

DELIVERING MICROSERVICES

RELEASE N - SCALE



DELIVERING MICROSERVICES

RELEASE N - SCALING FDELIVERY

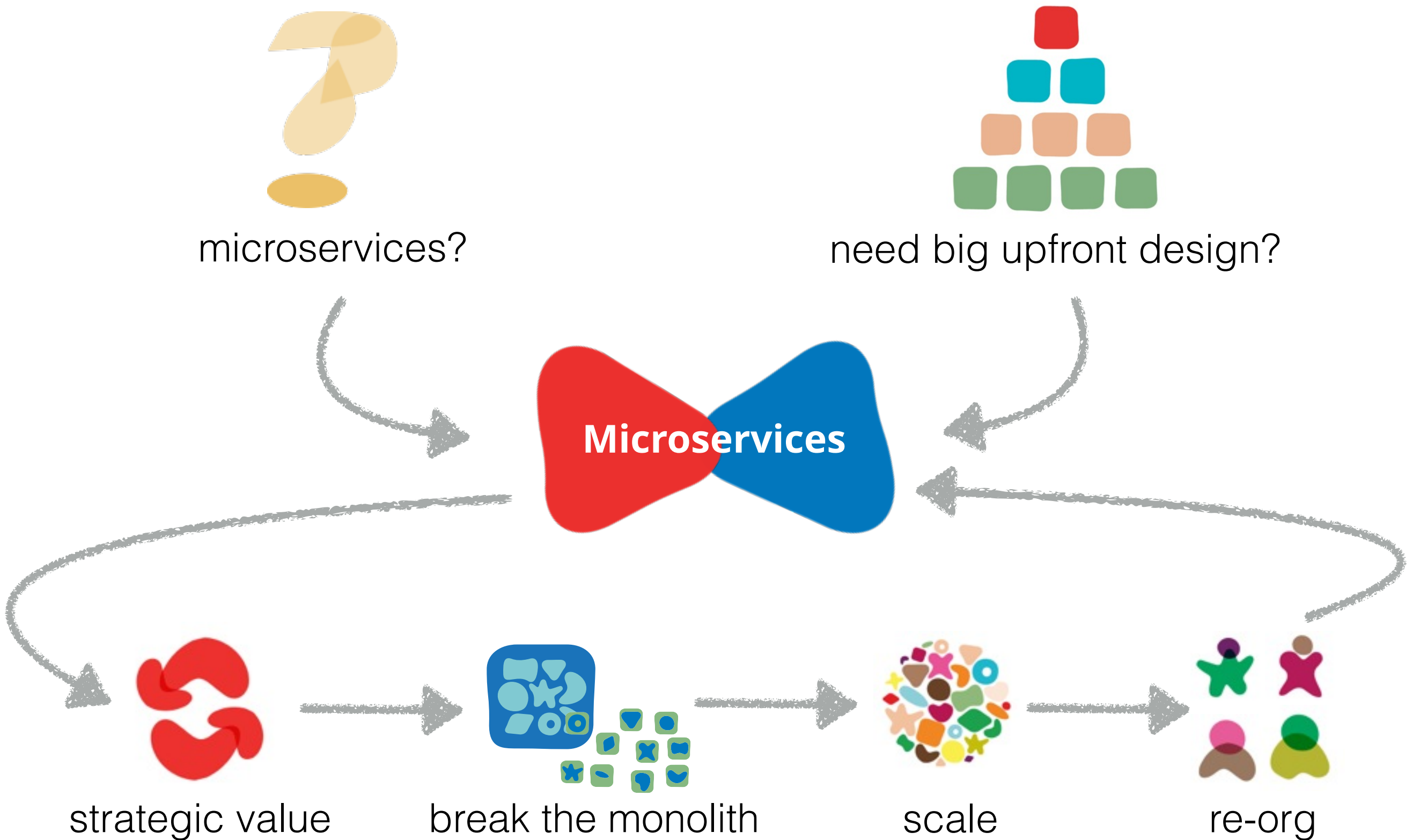
SCALABLE
ARCHITECTURE ??

DELIVERING MICROSERVICES

RELEASE N - SCALING FOR LOAD

DISTRIBUTING THE
TEAMS?

CONCLUSION



ABOUT ME

Jean D'Amore

Technology Market Principal

ThoughtWorks®



jdamore@thoughtworks.com



[jeandamore](#)



[jeandamore](#)



[jdamore](#)



blog.corsamore.com