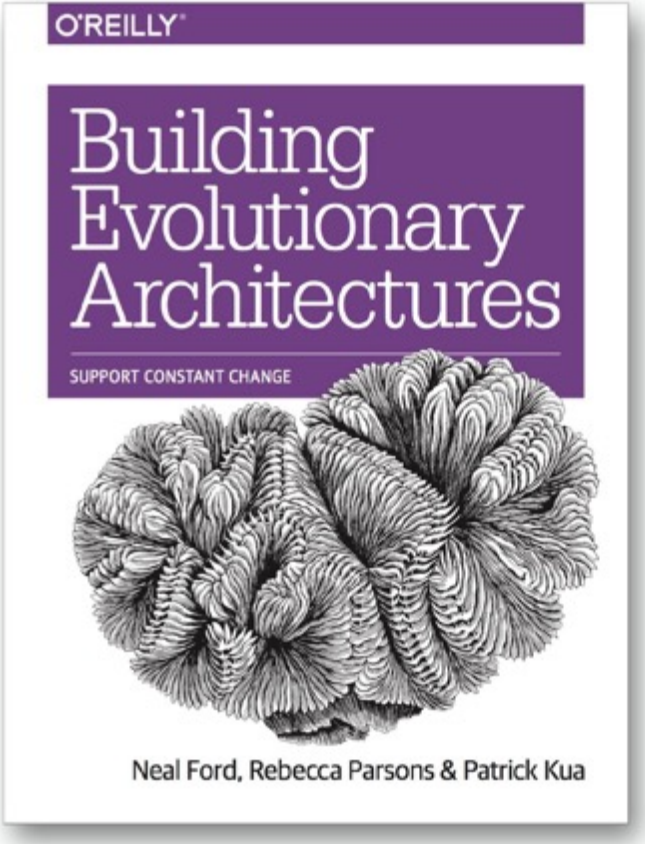
iMpLemENtiNg eVoLuTiONaRy ARcHitECtuREs



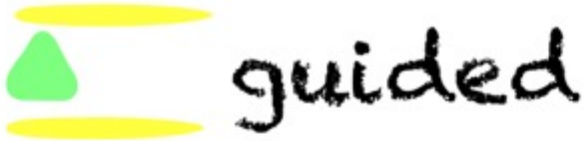
@neal4d
nealford.com

@rebeccaparsons

@patkua

Evolutionary Architecture

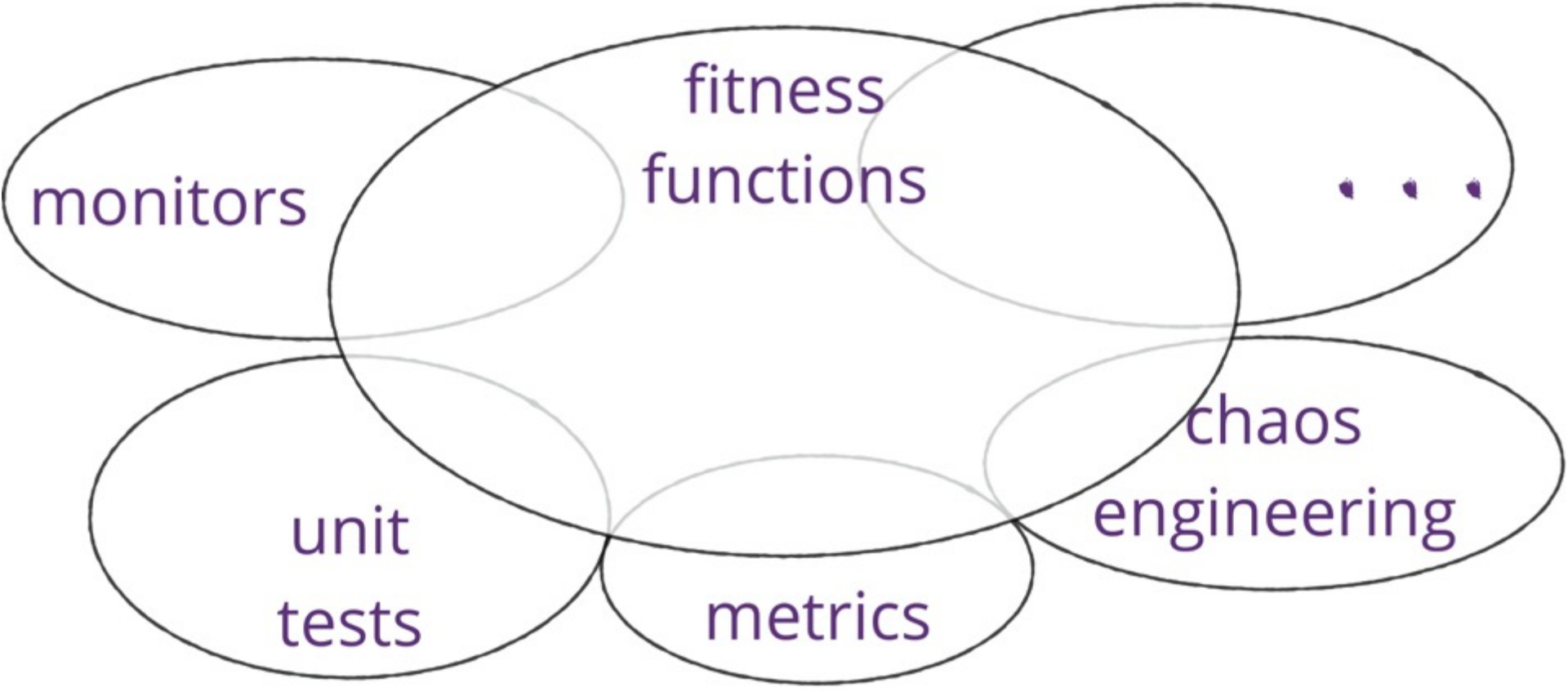
An evolutionary architecture supports
guided,
incremental change
across multiple dimensions.



***architectural** fitness function*:

An architectural fitness function provides an objective integrity assessment of some architectural characteristic(s).

Fitness Functions



Exercise:

What type of fitness function?

evolutionary architecture

governance

process

enterprise architecture

Agenda Part 2

The Evolution of Penultimate

Agenda Part 2

fitness functions

architecture migration

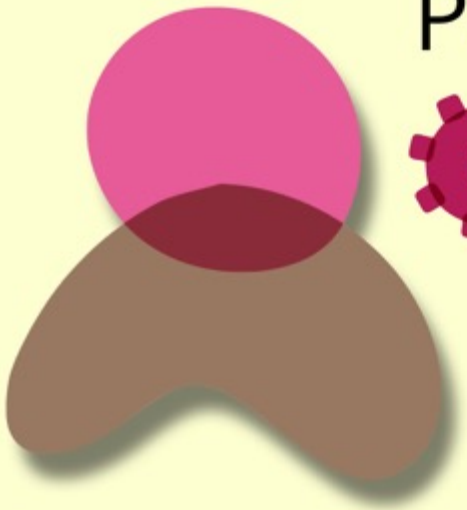
The Evolution of Penultimate

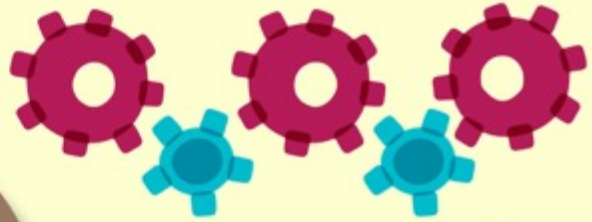
experimentation

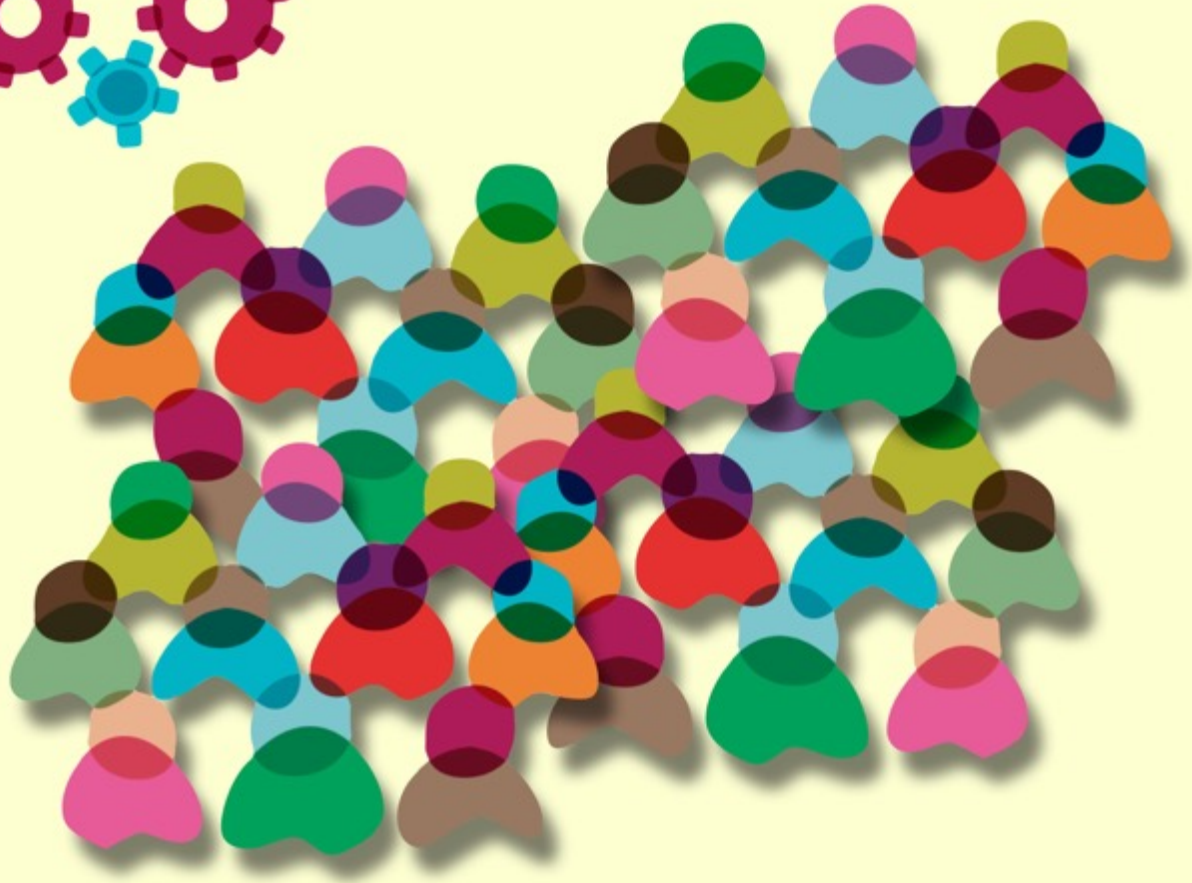
automating governance

hypothesis/data driven development

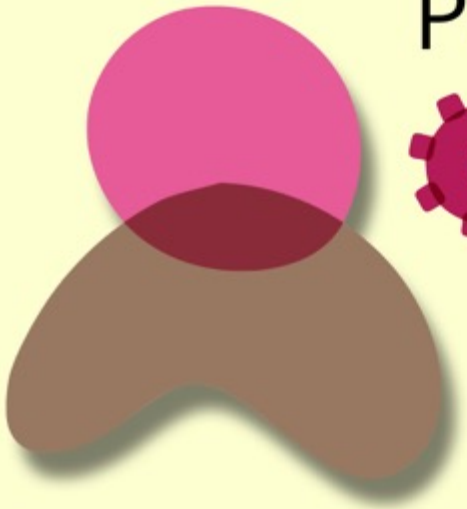
Penultima e

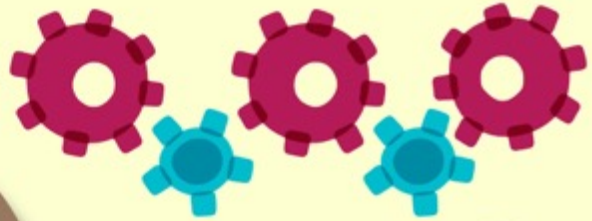


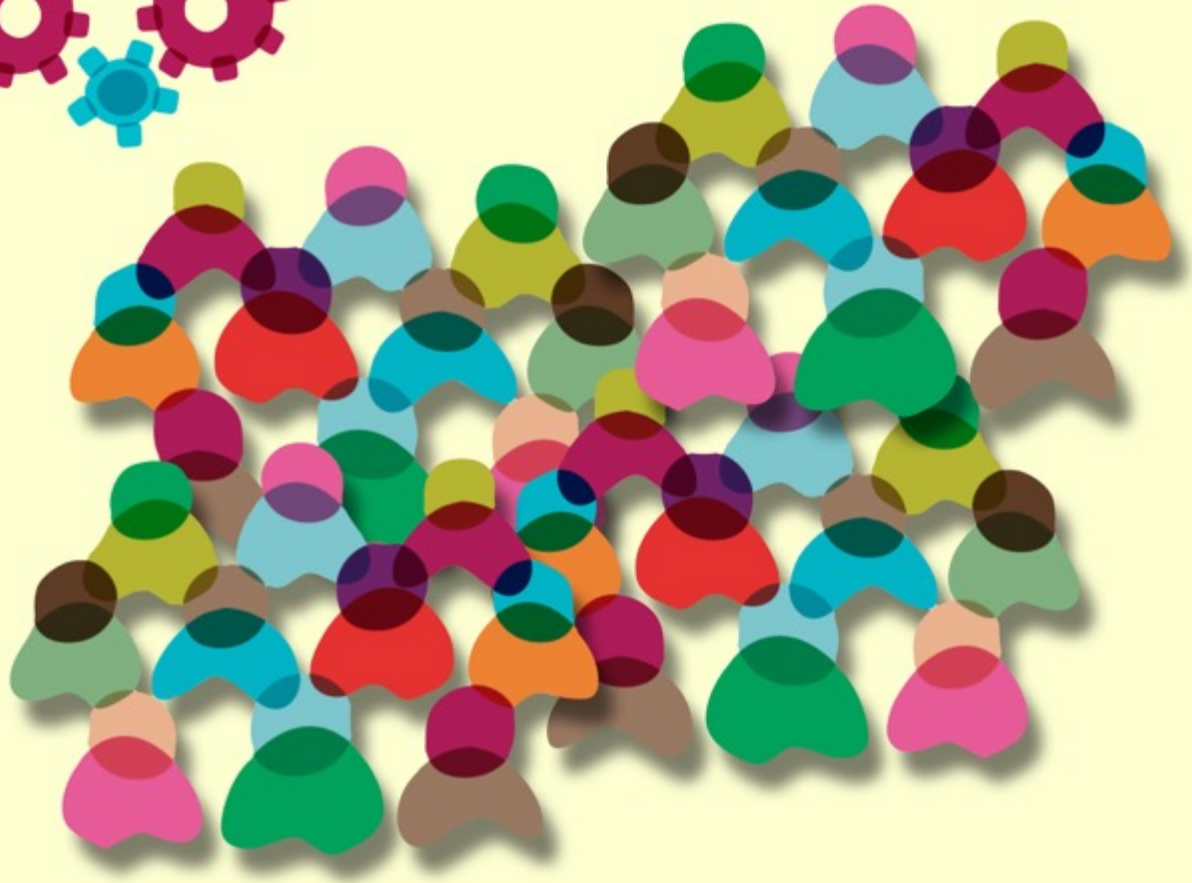




Penultimate





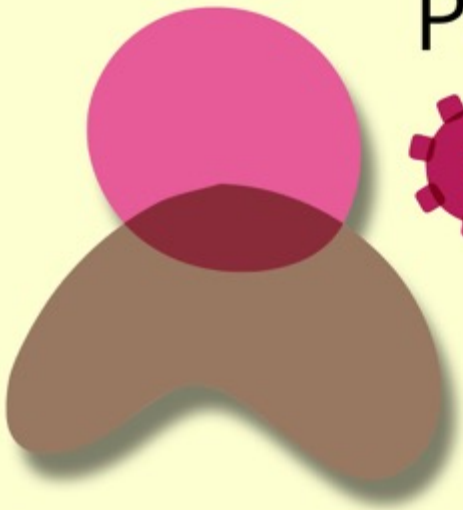


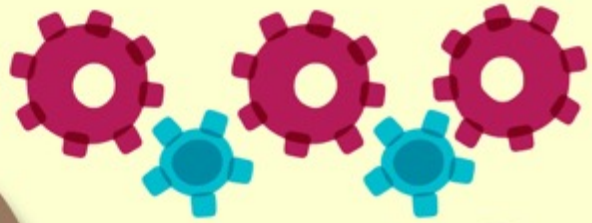
pe·nul·ti·mate
/pəˈnəltəmət/

adjective

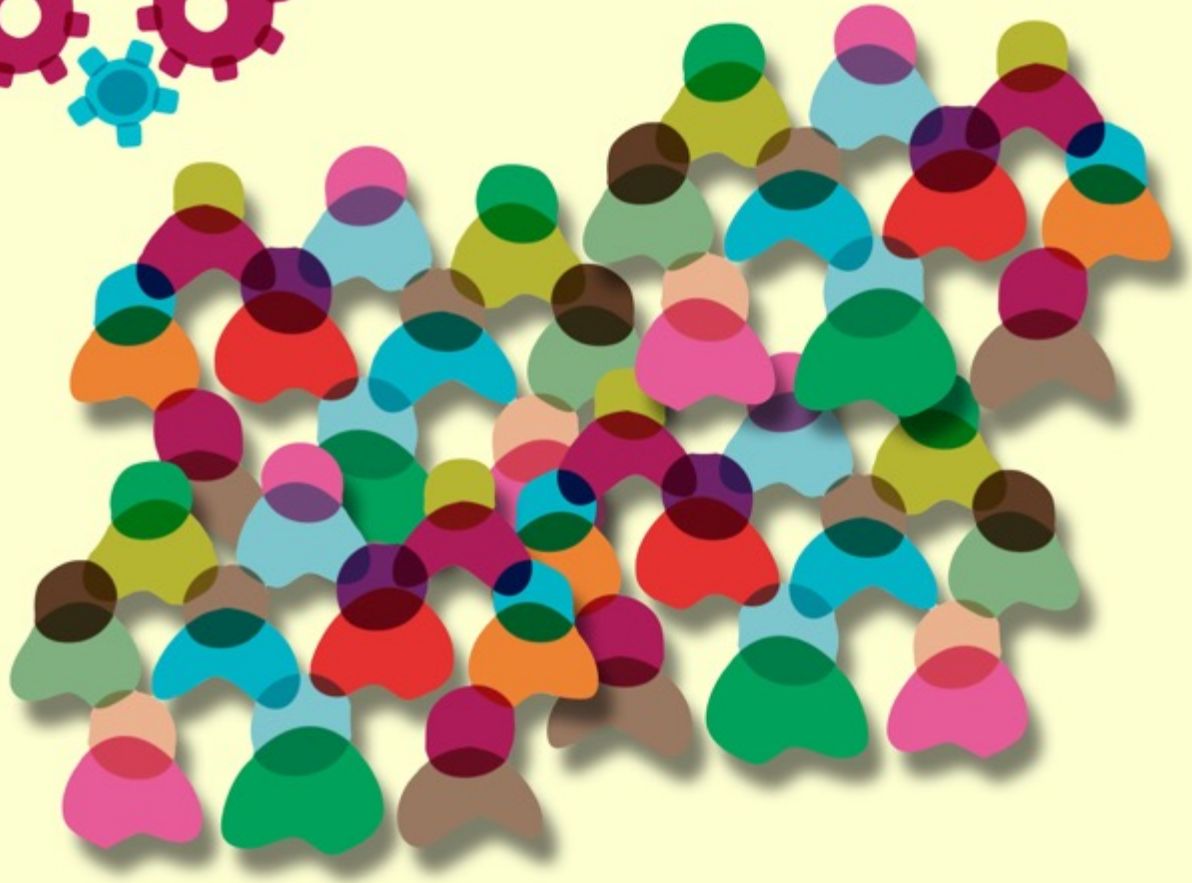
last but one in a series of things; second to the last.
"the penultimate chapter of the book"
synonyms: next-to-last, second-to-last, second-last
"the penultimate movie on my top-ten list is an animated feature"

Penultimate

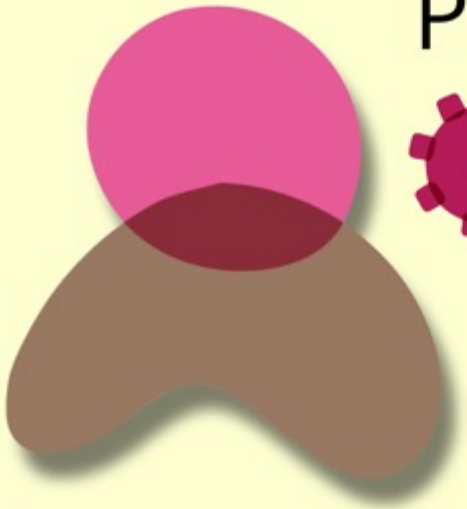




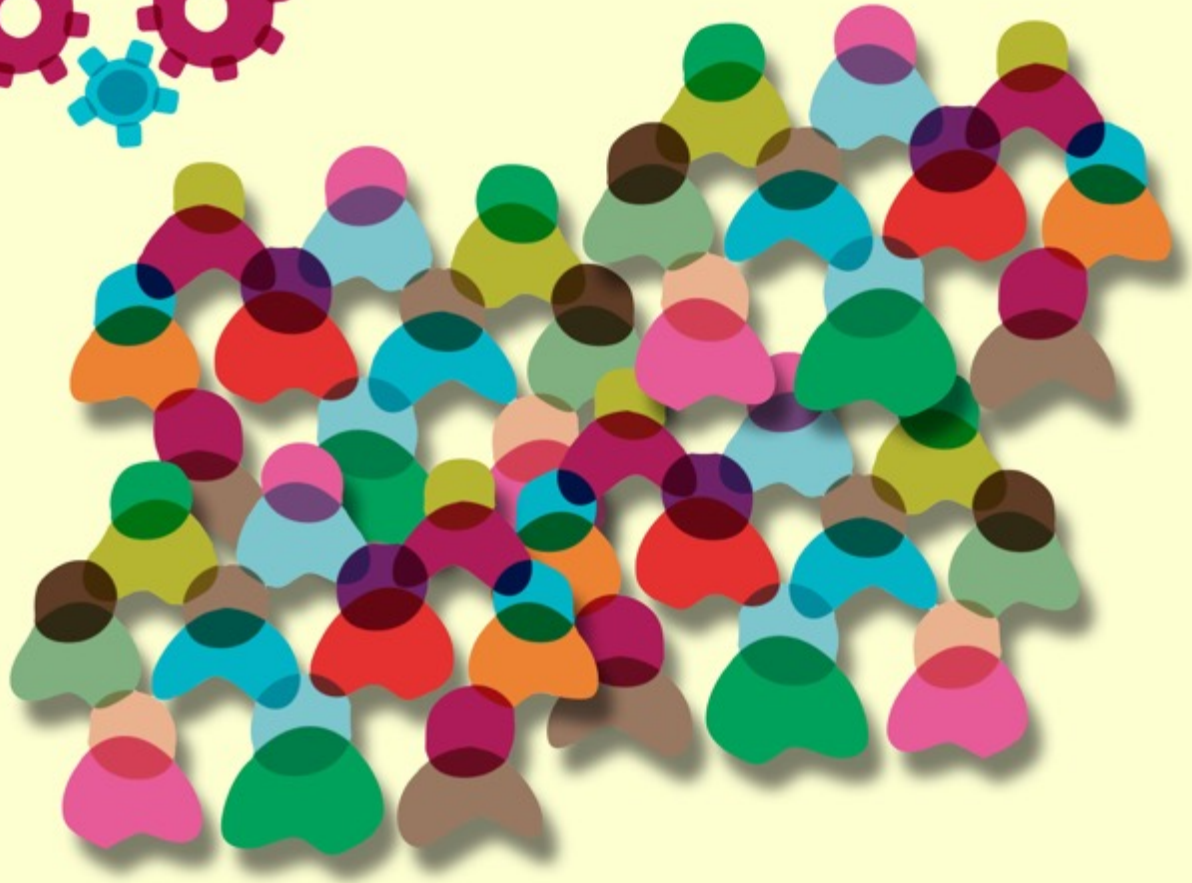
— many outdated systems



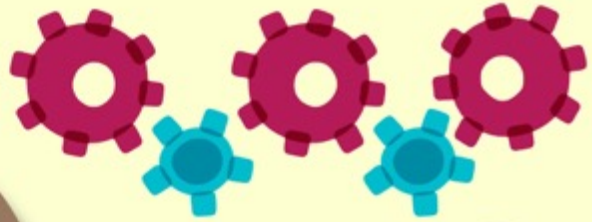
Penultimate

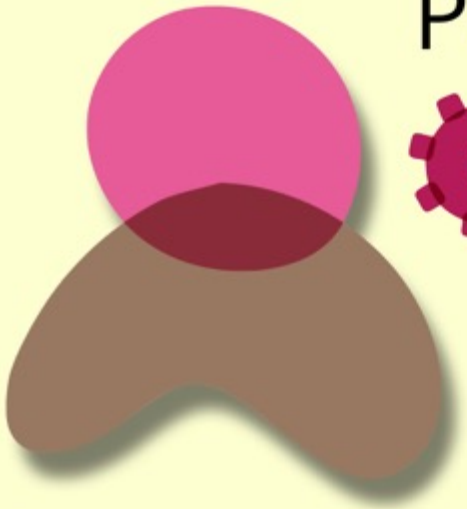


— many outdated systems
— outsourced operations

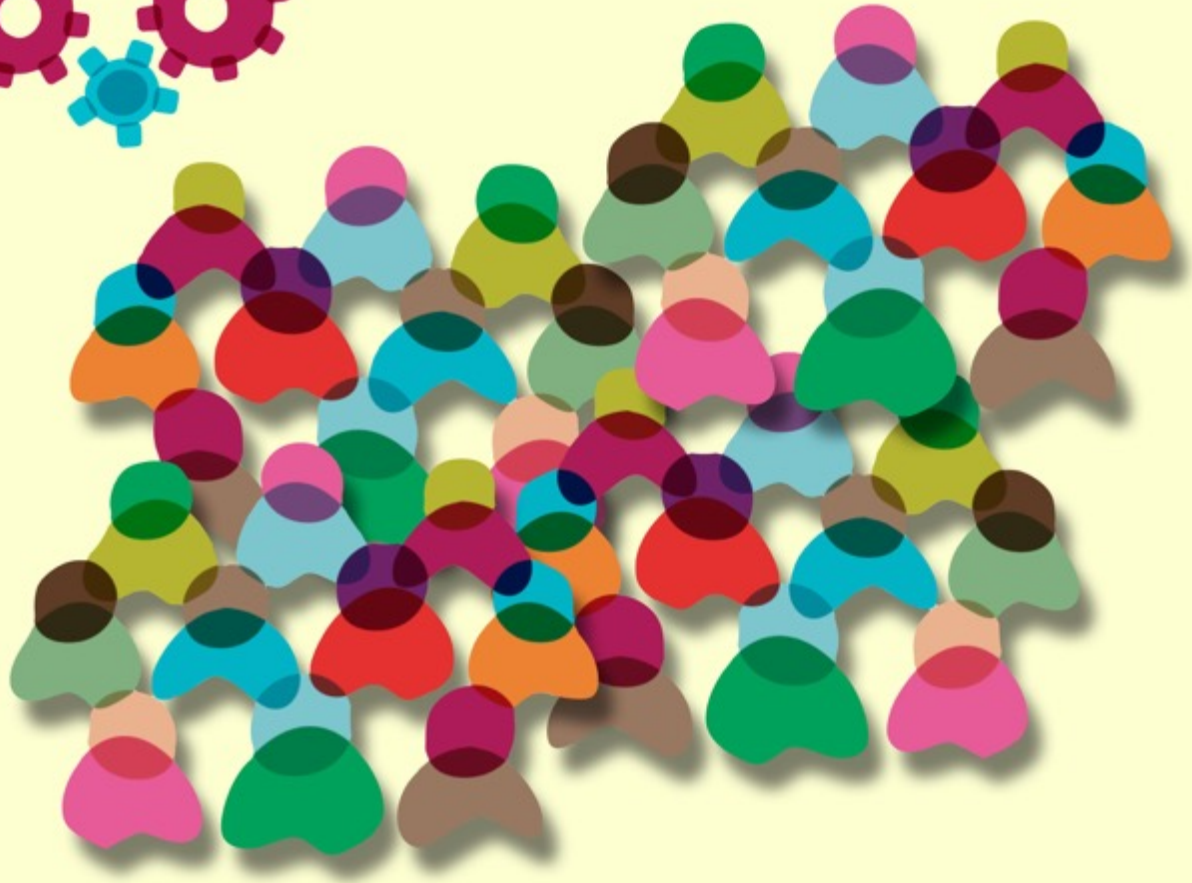


Penultimate

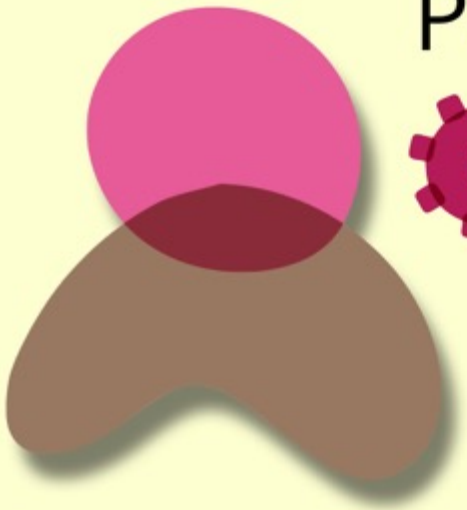


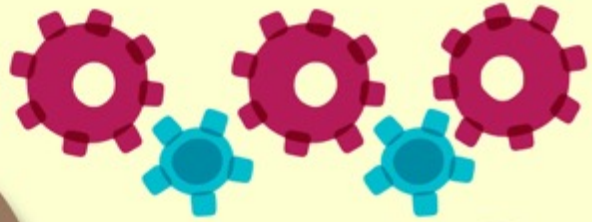


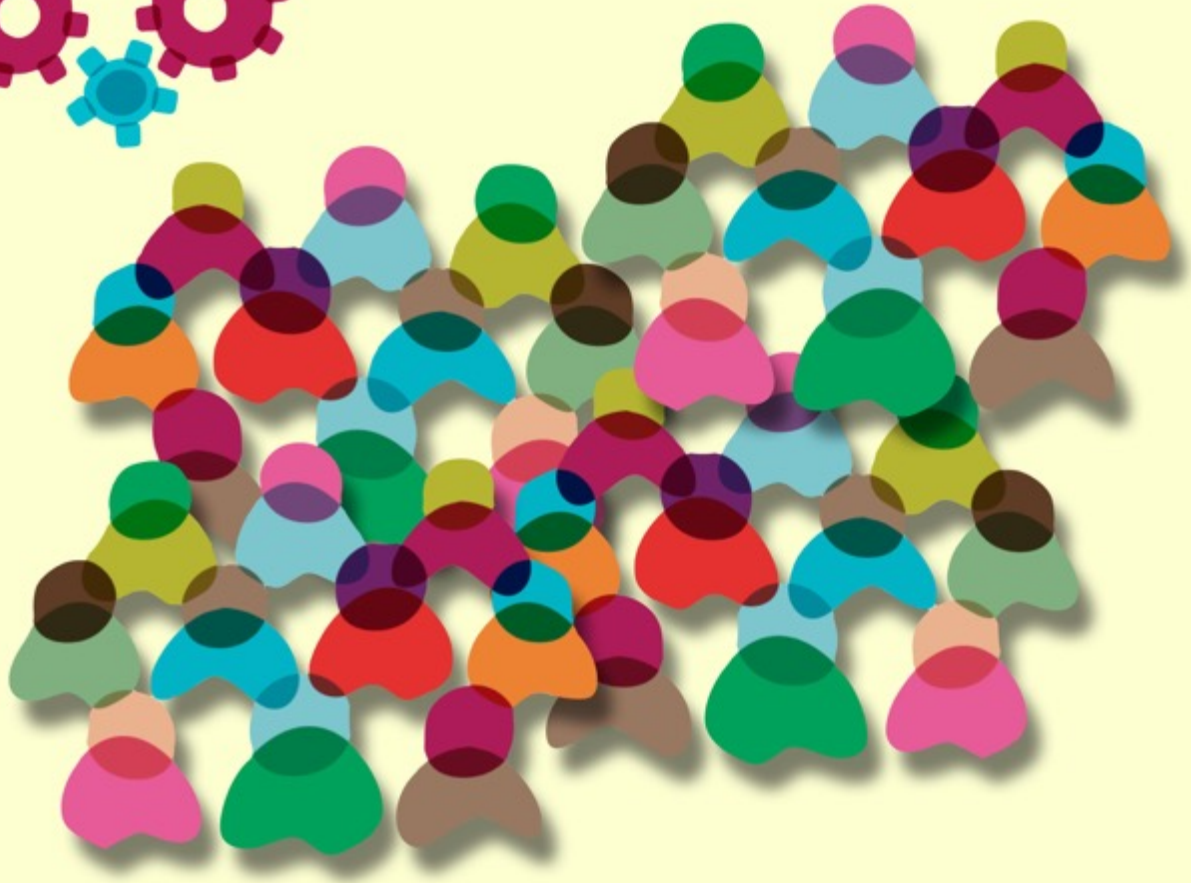
— many outdated systems
— outsourced operations
— impending competitors



Penultimate

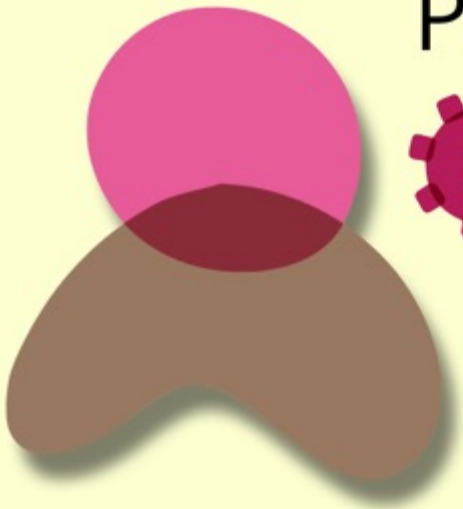


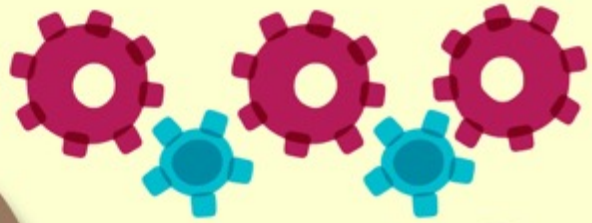


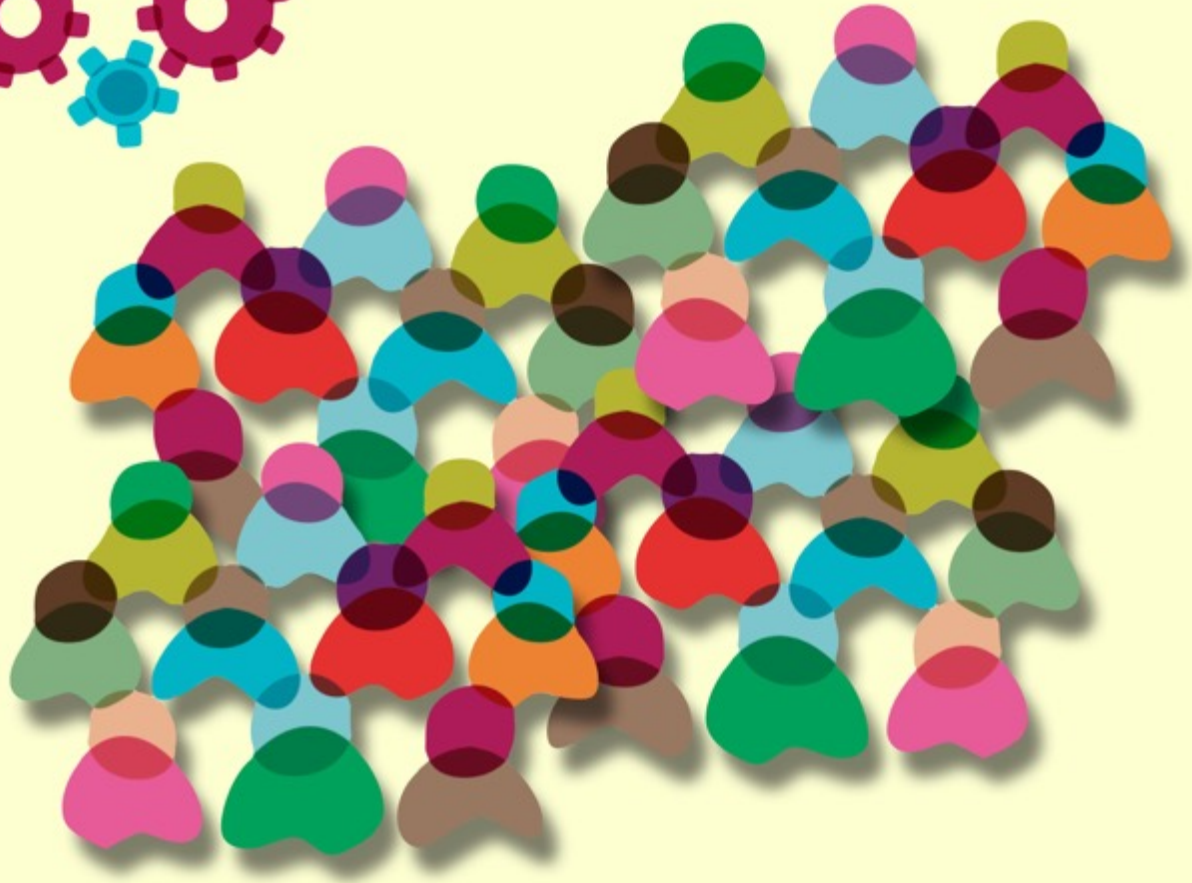


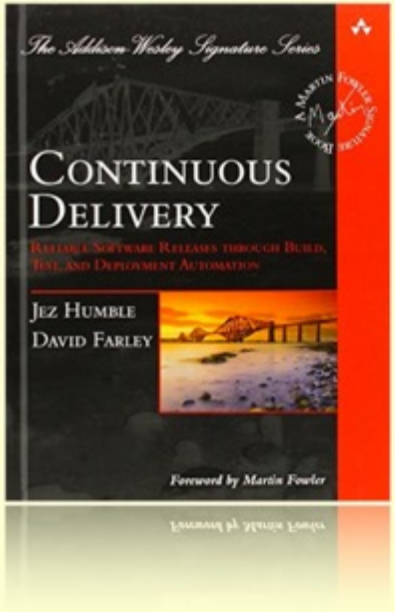
— many outdated systems
— outsourced operations
— impending competitors
— second to last in market!

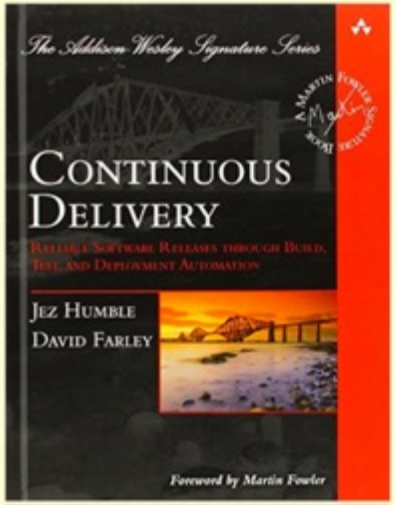
Penultima e

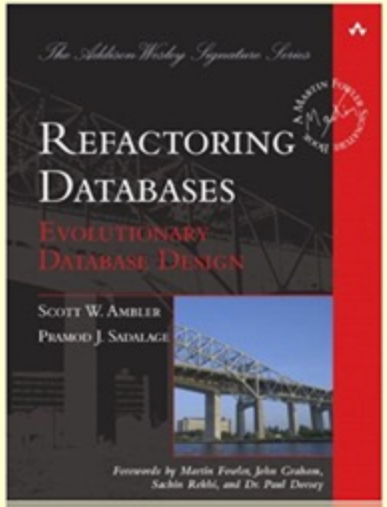


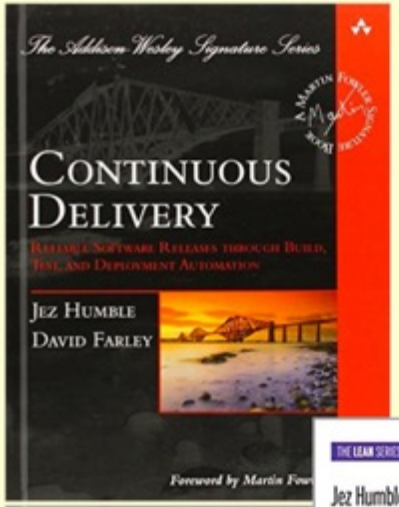


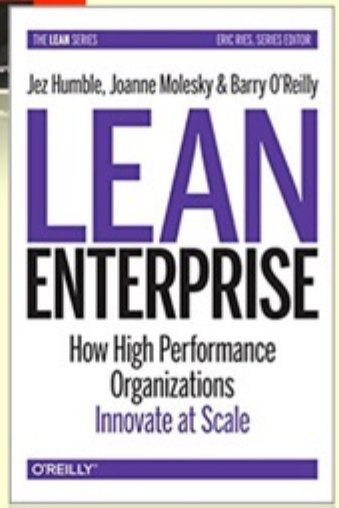


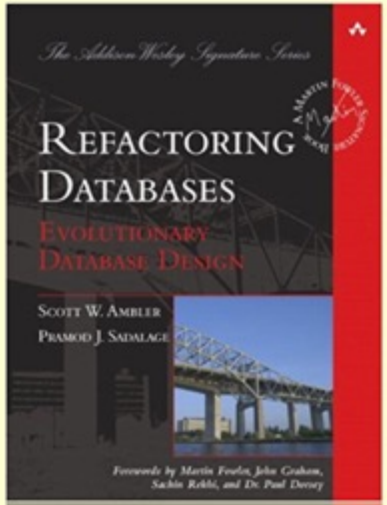

The Addison-Wesley Signature Series
A Martin Fowler Signature Book
Continuous Delivery
Reliable Software Releases through Build, Test, and Deployment Automation
Jez Humble
David Farley
Foreword by Martin Fowler

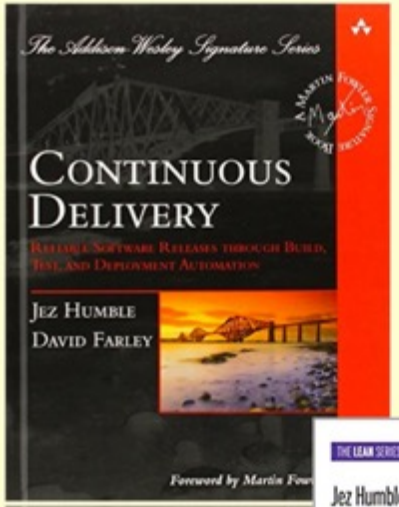

The Addison Wesley Signature Series
A Martin Fowler Signature Book
Continuous Delivery
Reliable Software Releases through Build, Test, and Deployment Automation
Jez Humble
David Farley
Foreword by Martin Fowler

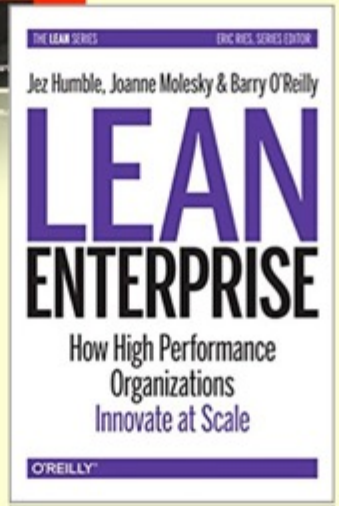

The Addison Wesley Signature Series
A Martin Fowler Signature Book
Refactoring Databases
Evolutionary Database Design
Scott W. Ambler
Pramod J. Sadalage
Forewords by Martin Fowler, John Graham, Sachin Rekhi, and Dr. Paul Dorsey

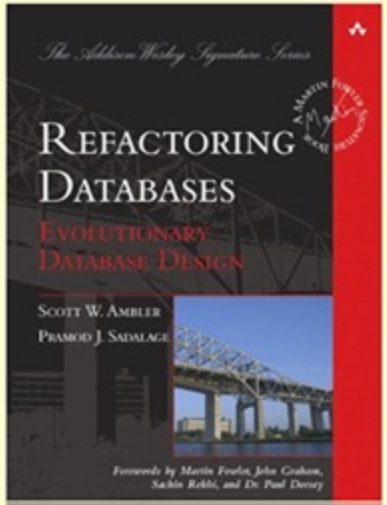

The Addison Wesley Signature Series
CONTINUOUS DELIVERY
Reliable Software Releases through Build, Test, and Deployment Automation
Jez Humble
David Farley
Foreword by Martin Fowler


THE LEAN SERIES
ERIC RIES, SERIES EDITOR
Jez Humble, Joanne Molesky & Barry O'Reilly
LEAN ENTERPRISE
How High Performance Organizations Innovate at Scale
O'REILLY

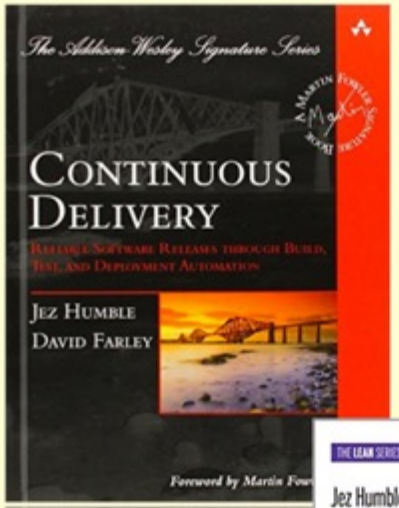

The Addison Wesley Signature Series
REFACTORING DATABASES
Evolutionary Database Design
Scott W. Ambler
Pramod J. Sadalage
Forewords by Martin Fowler, John Graham, Sachin Rekhi, and Dr. Paul Dorsey

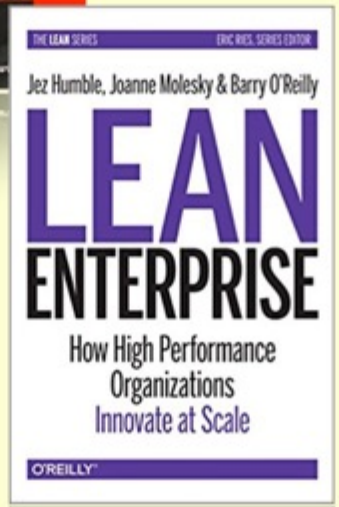

The Addison-Wesley Signature Series
Continuous Delivery
Reliable Software Releases through Build, Test, and Deployment Automation
Jez Humble
David Farley
Foreword by Martin Fowler

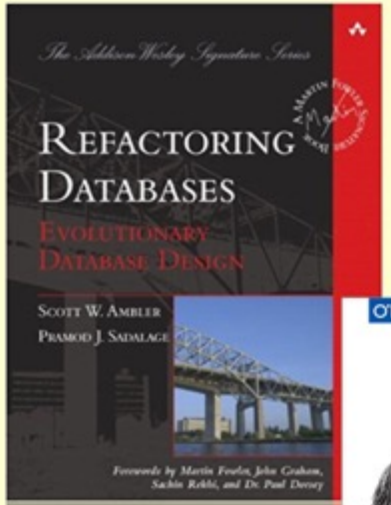

The Lean Series
Eric Ries, Series Editor
Jez Humble, Joanne Molesky & Barry O'Reilly
Lean Enterprise
How High Performance Organizations Innovate at Scale
O'Reilly

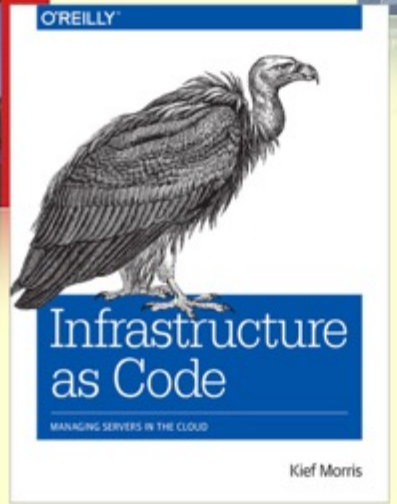

The Addison-Wesley Signature Series
Refactoring Databases
Evolutionary Database Design
Scott W. Ambler
Pramod J. Sadalage
Forewords by Martin Fowler, John Graham, Sachin Rekhi, and Dr. Paul Dorsey


The Pragmatic Programmers
Release It!
Second Edition
Design and Deploy Production-Ready Software
Michael T. Nygard
Edited by Katherine Dvorak

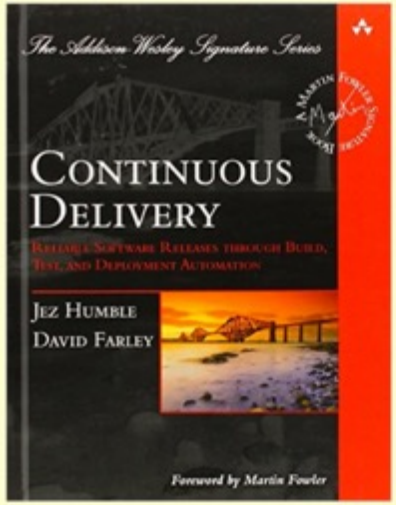

The Addison-Wesley Signature Series
Continuous Delivery
Reliable Software Releases through Build, Test, and Deployment Automation
Jez Humble
David Farley
Foreword by Martin Fowler

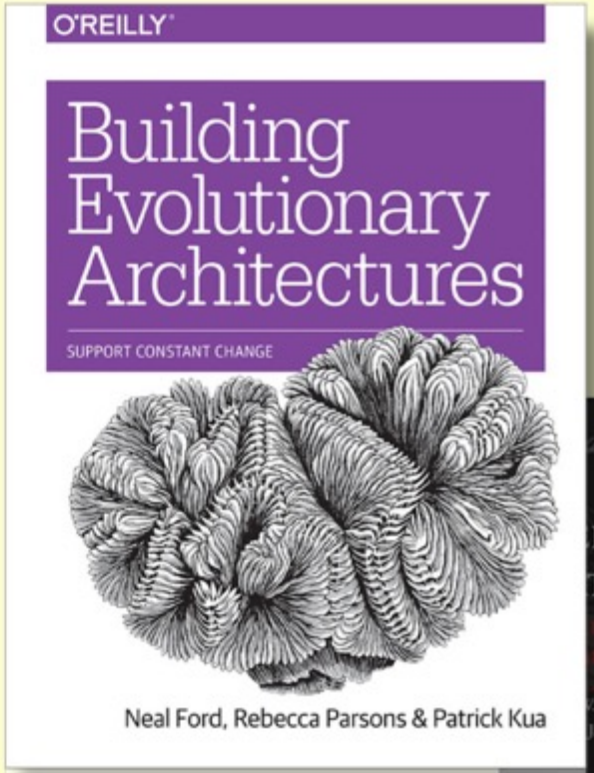

The Lean Series
Eric Ries, Series Editor
Jez Humble, Joanne Molesky & Barry O'Reilly
Lean Enterprise
How High Performance Organizations Innovate at Scale
O'Reilly


The Addison-Wesley Signature Series
Refactoring Databases
Evolutionary Database Design
Scott W. Ambler
Pramod J. Sadalage
Forewords by Martin Fowler, John Graham, Sachin Rekhi, and Dr. Paul Dorsey

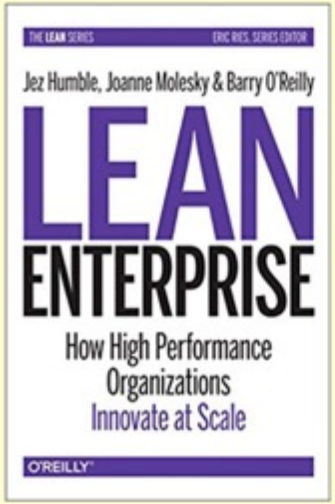

O'Reilly
Infrastructure as Code
Managing Servers in the Cloud
Kief Morris

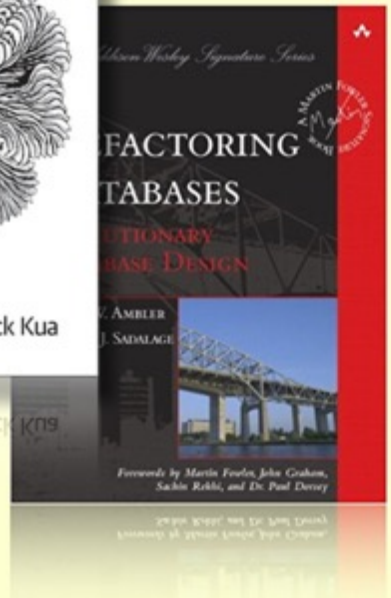

The Pragmatic Programmers
Release It!
Second Edition
Design and Deploy Production-Ready Software
Michael T. Nygard

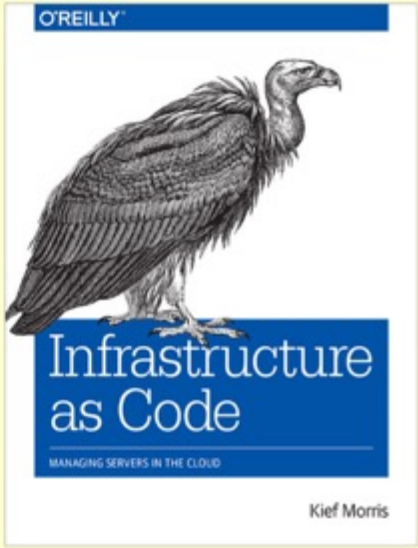

The Addison Wesley Signature Series
CONTINUOUS DELIVERY
Reliable Software Releases through Build, Test, and Deployment Automation
JEZ HUMBLE
DAVID FARLEY
Foreword by Martin Fowler


O'REILLY
Building Evolutionary Architectures
SUPPORT CONSTANT CHANGE
Neal Ford, Rebecca Parsons & Patrick Kua

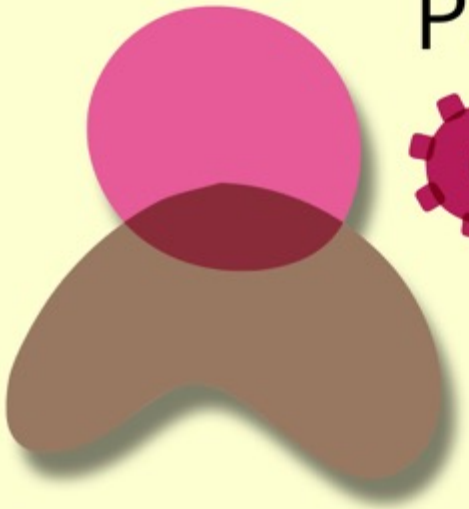

The Pragmatic Programmers
Release It!
Second Edition
Design and Deploy Production-Ready Software
Michael T. Nygard


THE LEAN SERIES
ERIC RIES, SERIES EDITOR
Jez Humble, Joanne Molesky & Barry O'Reilly
LEAN ENTERPRISE
How High Performance Organizations Innovate at Scale
O'REILLY


FACTORING
TABASES
Forewords by Martin Fowler, John Graham, Sachin Rekhi, and Dr. Paul Dovey

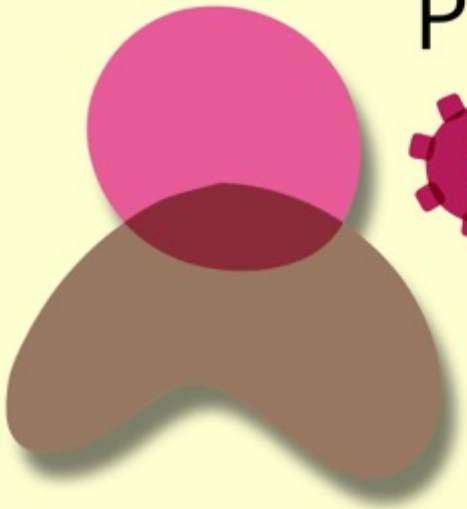

O'REILLY
Infrastructure as Code
MANAGING SERVERS IN THE CLOUD
Kief Morris

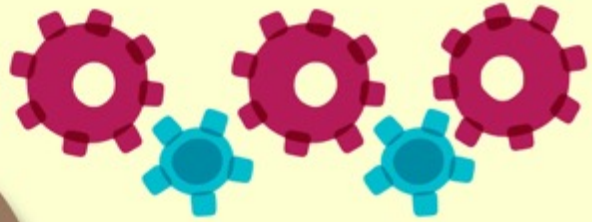
Penultimate



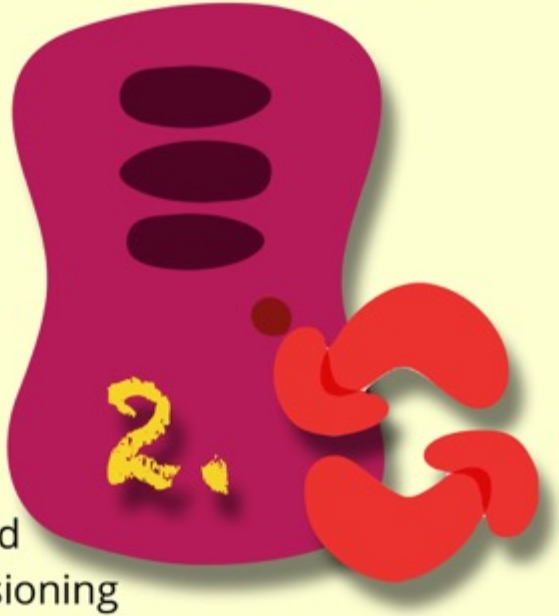
continuous integration/
deployment pipeline

Penultimate



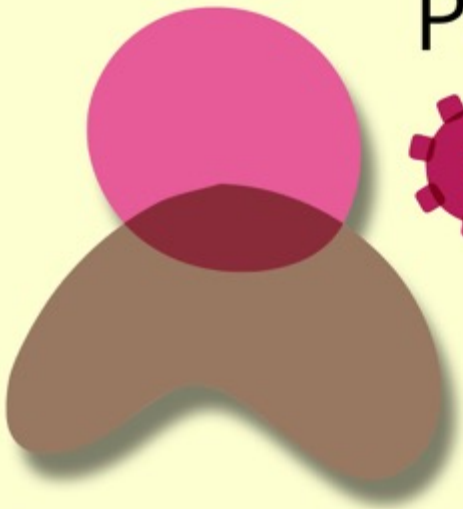


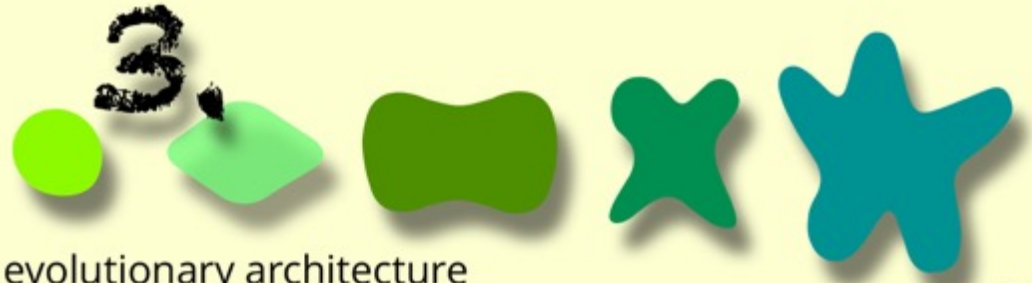
continuous integration/
deployment pipeline



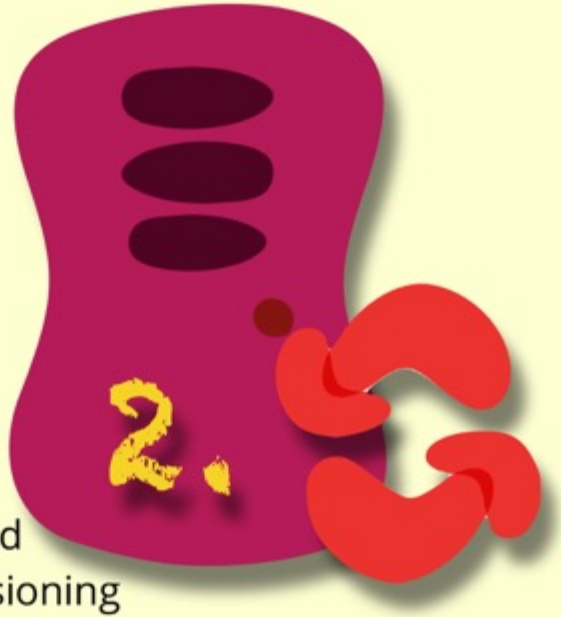
automated
machine provisioning

Penultimate



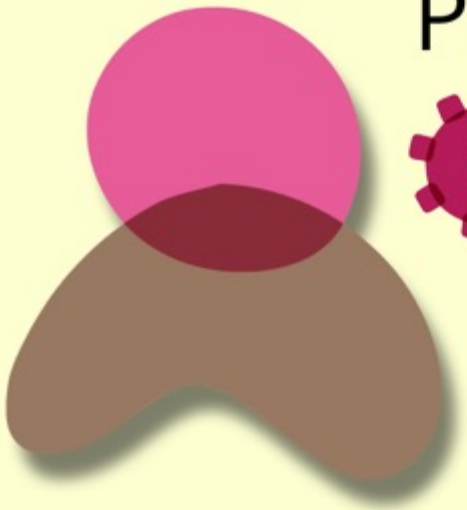


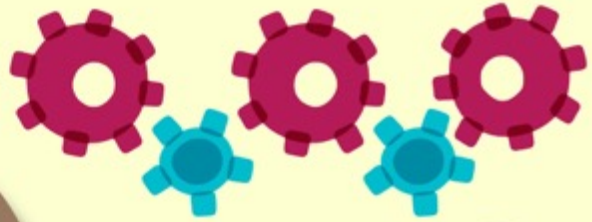
evolutionary architecture

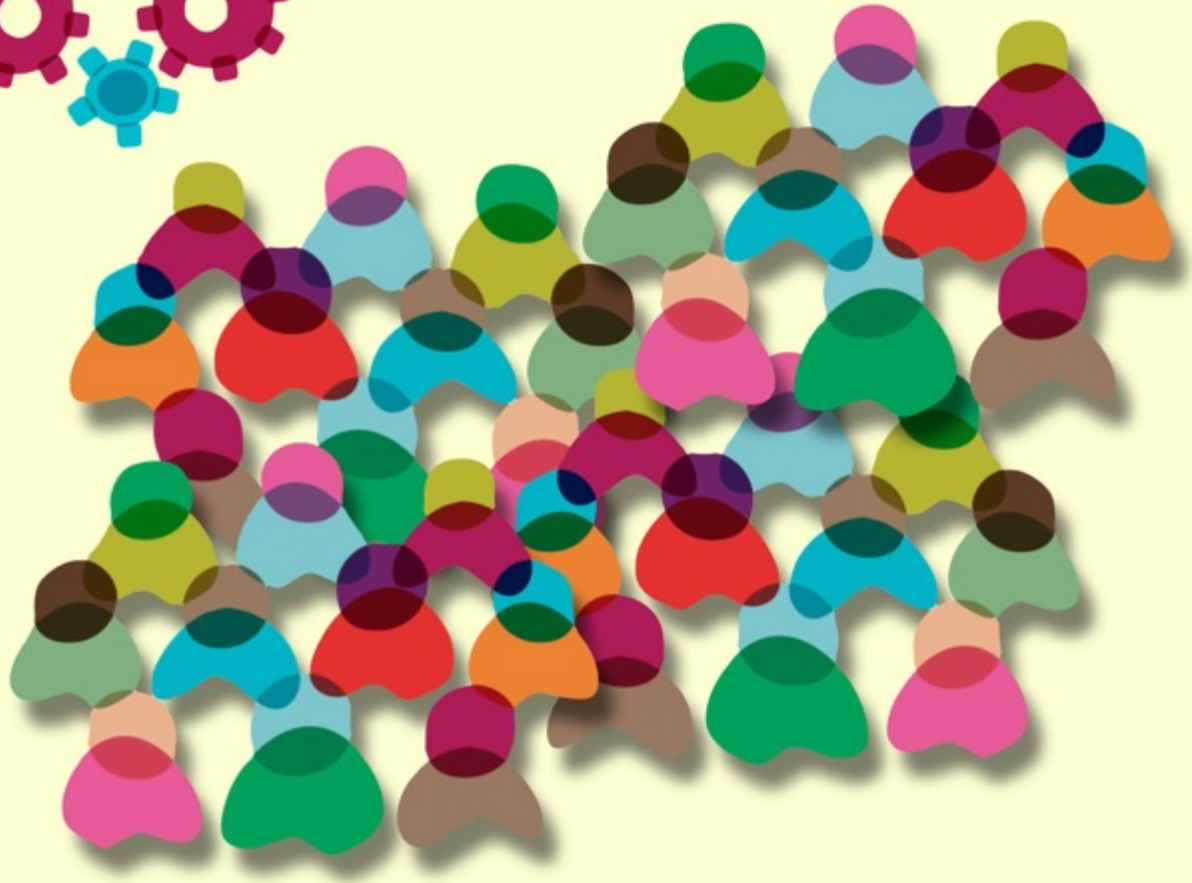


automated
machine provisioning

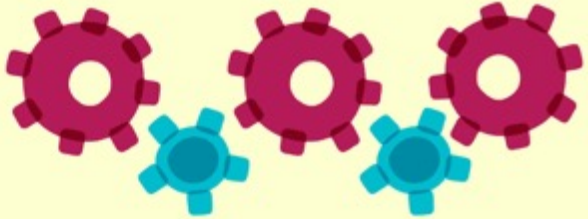
Penultima e

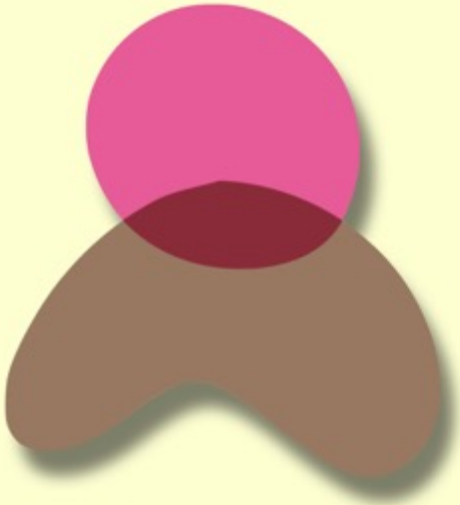






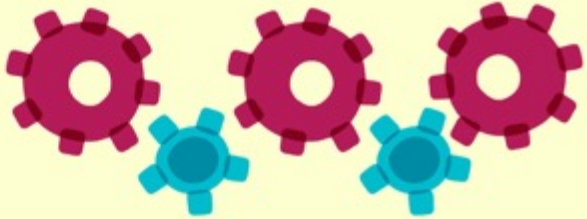
Penultima e

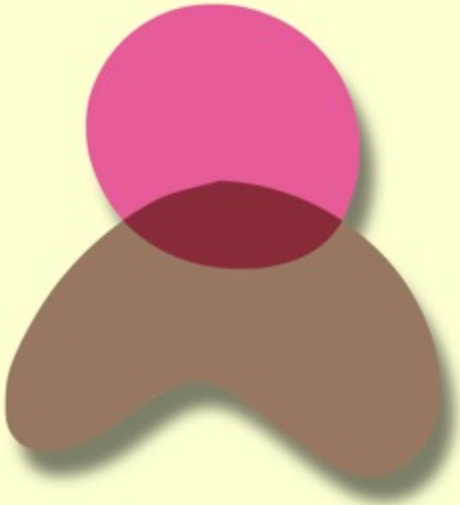




too complex!!

Penultimate

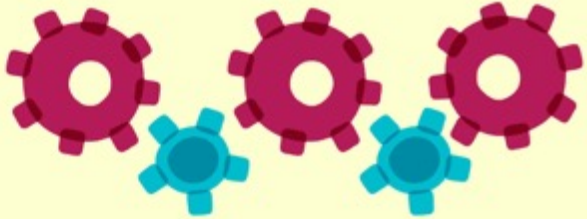


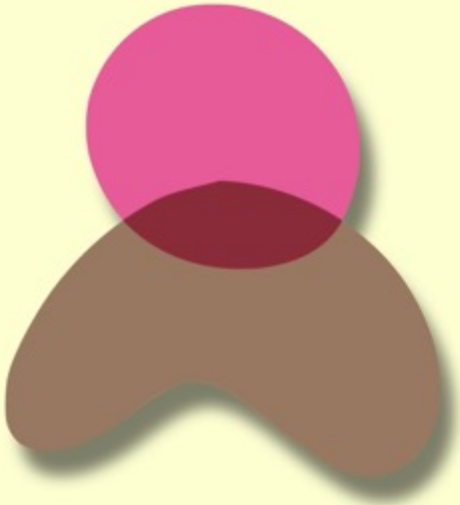


too complex!!

no expertise!!

Penultimate

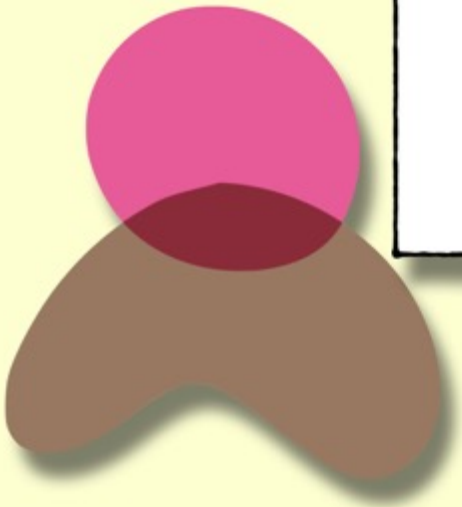




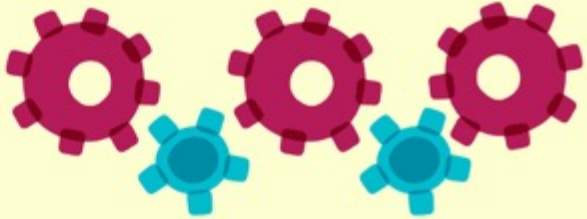
too complex!!

no expertise!!

where to start?!!



Penultimate

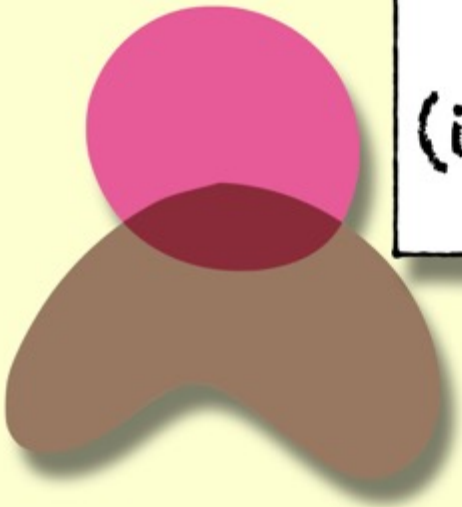


too complex!!

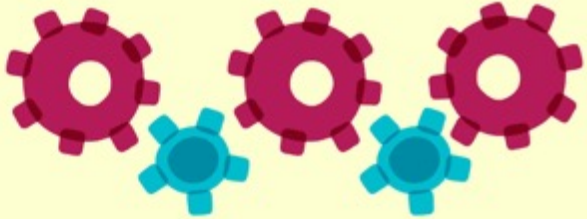
no expertise!!

where to start?!!

(it'll never work)



Penultimate

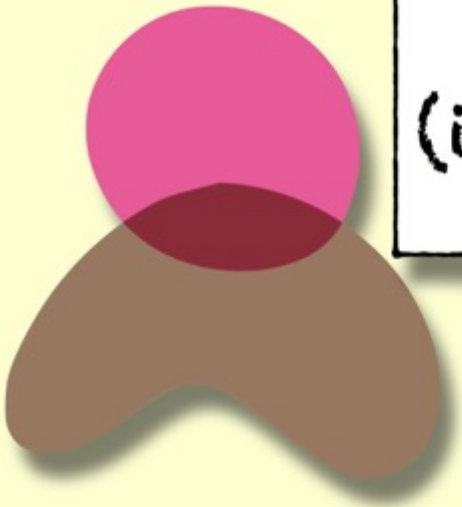


too complex!!

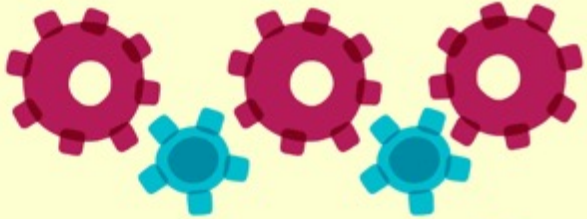
no expertise!!

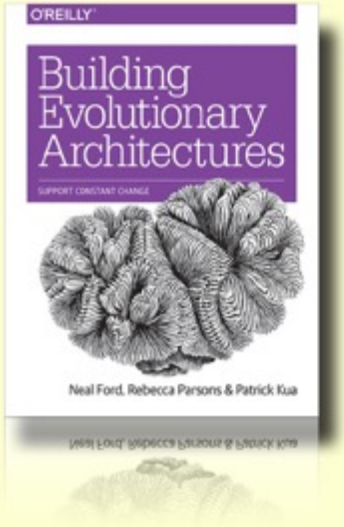
where to start?!!

(it'll never work)



Penultimate



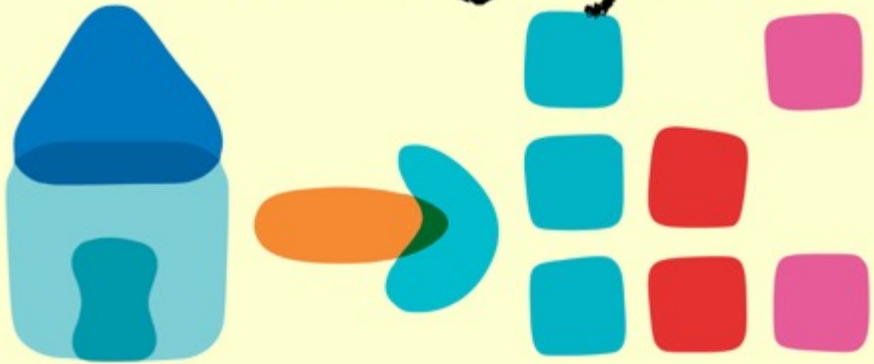


demonstration
defeats
discussion

Penultima e

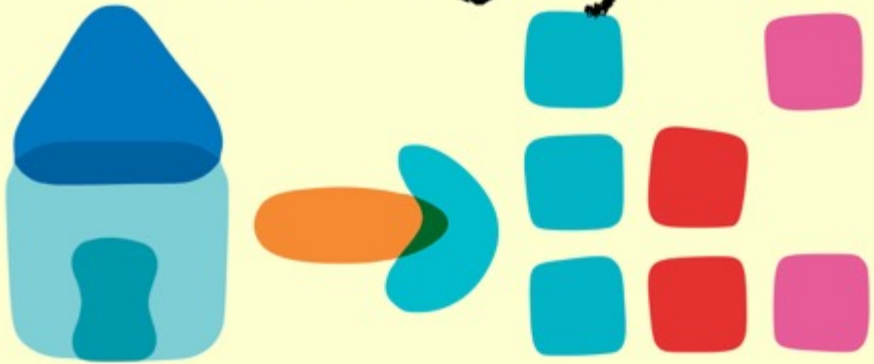
Penultimate

migrate critical existing systems

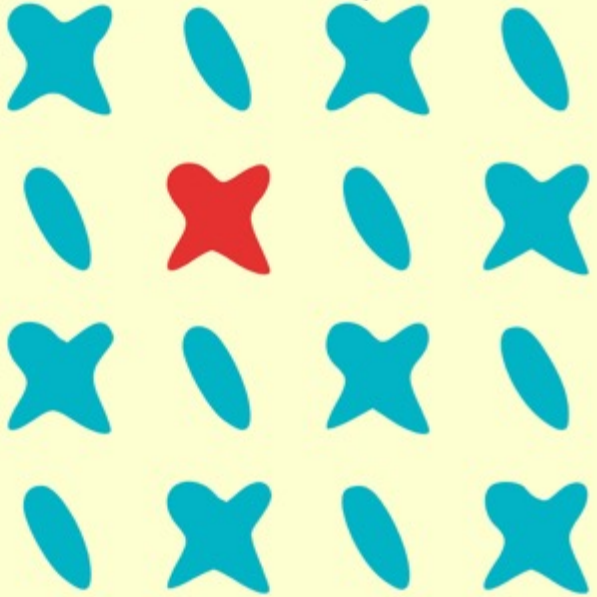


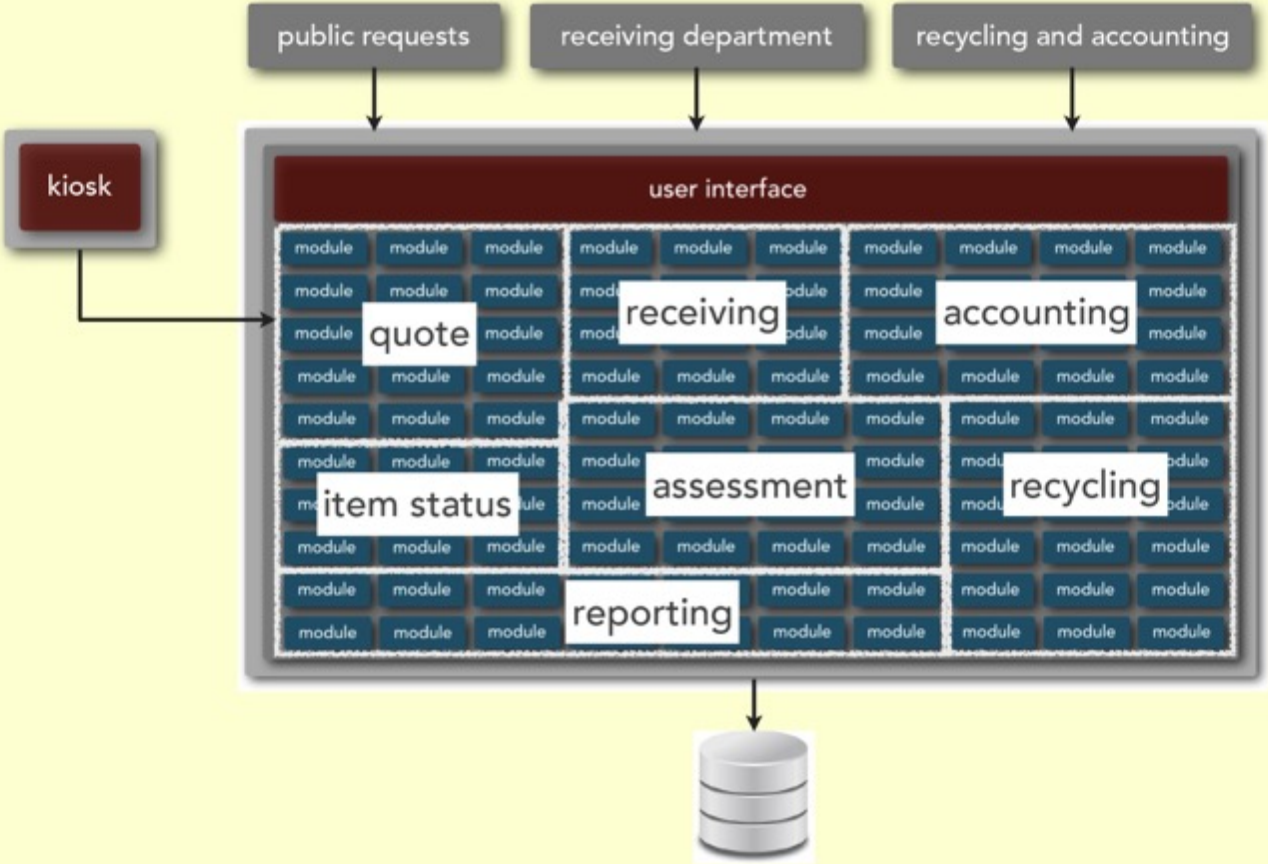
Penultimate

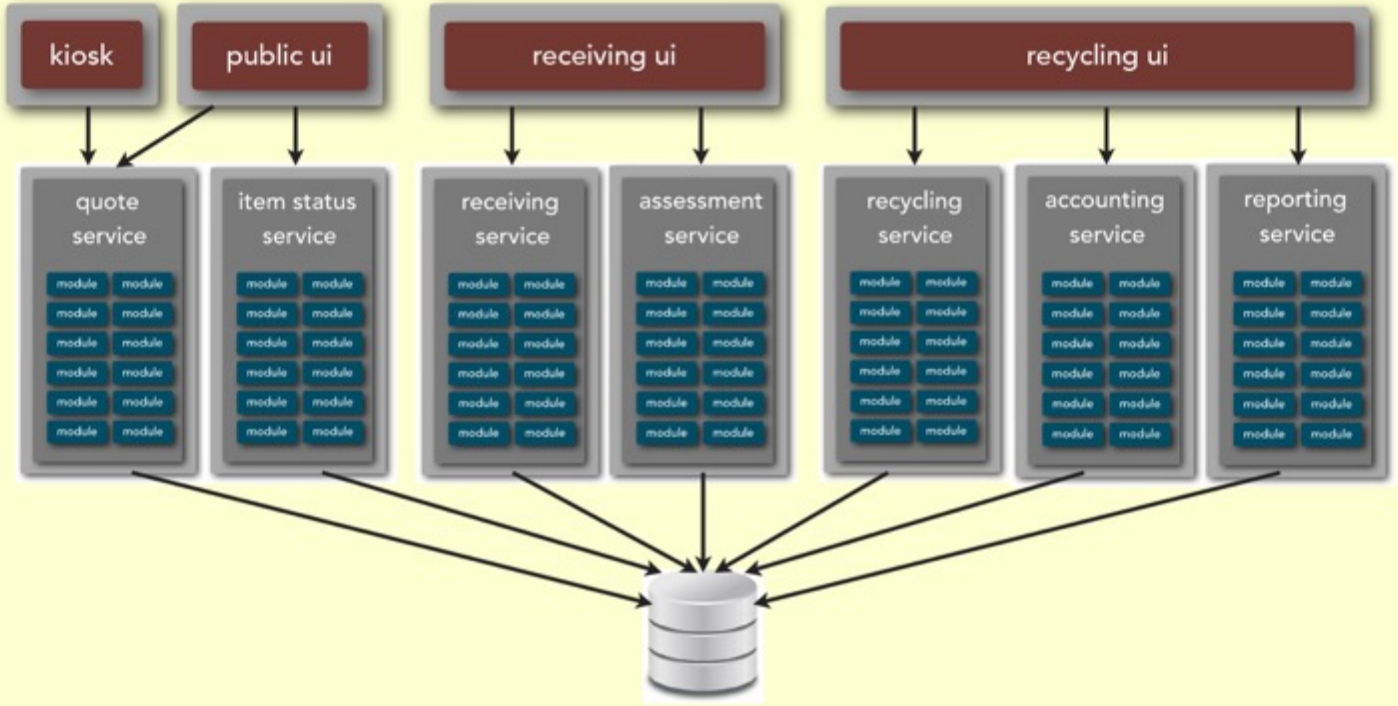
migrate critical existing systems



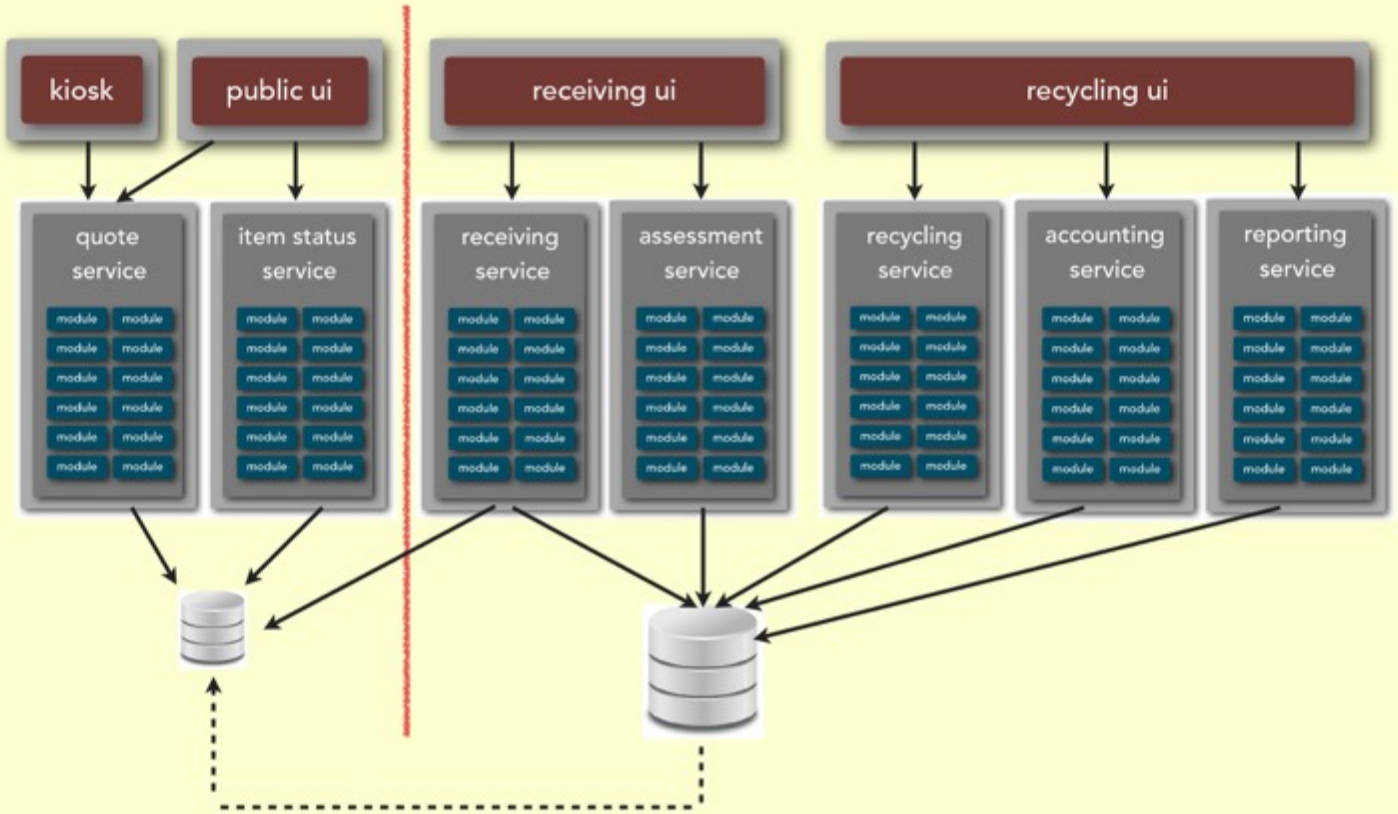
microservices for new development

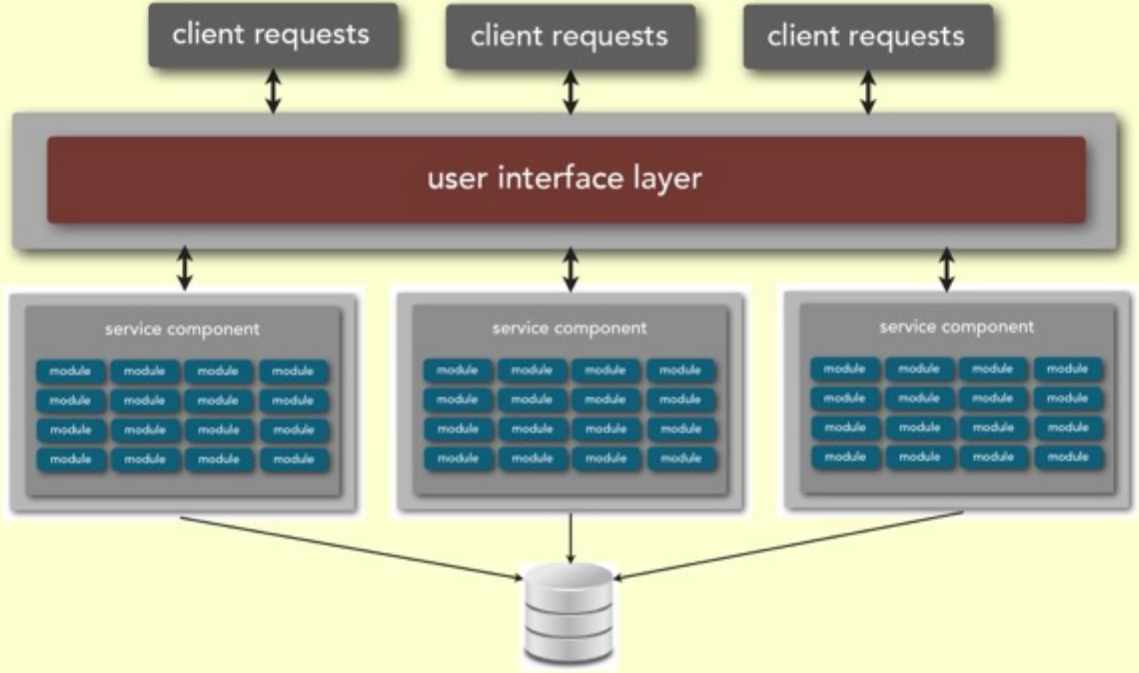


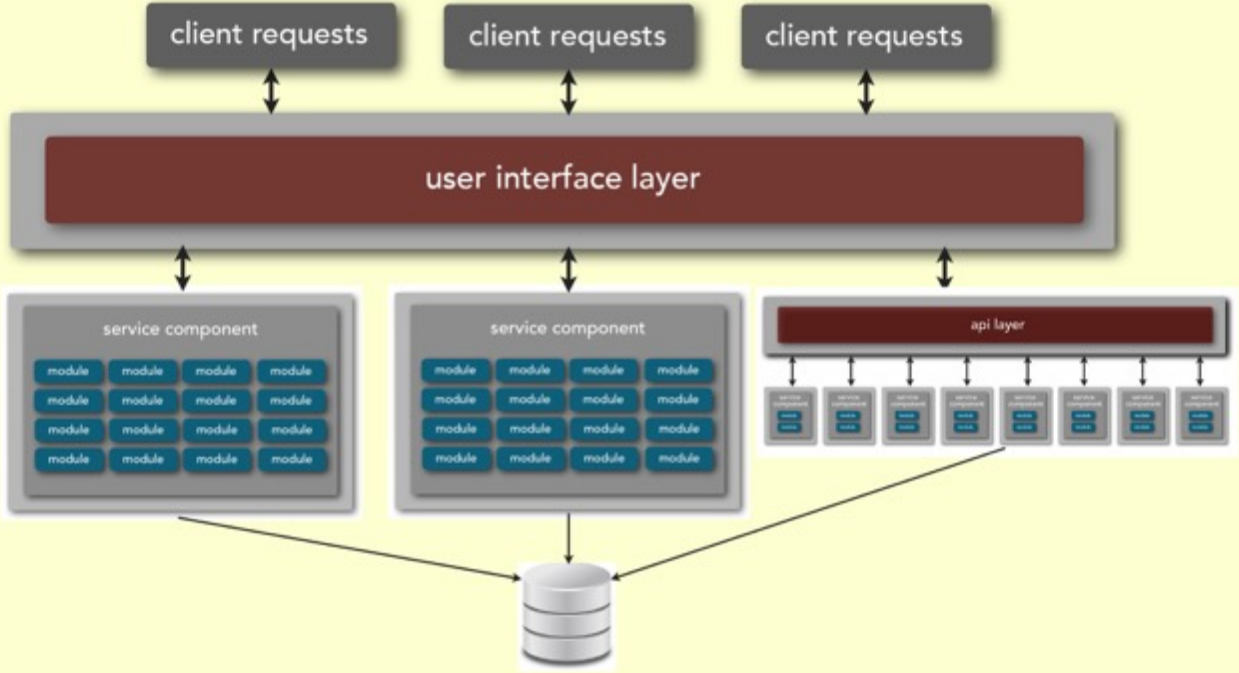

public requests
receiving department
recycling and accounting
kiosk
user interface
quote
receiving
accounting
item status
assessment
recycling
reporting
module

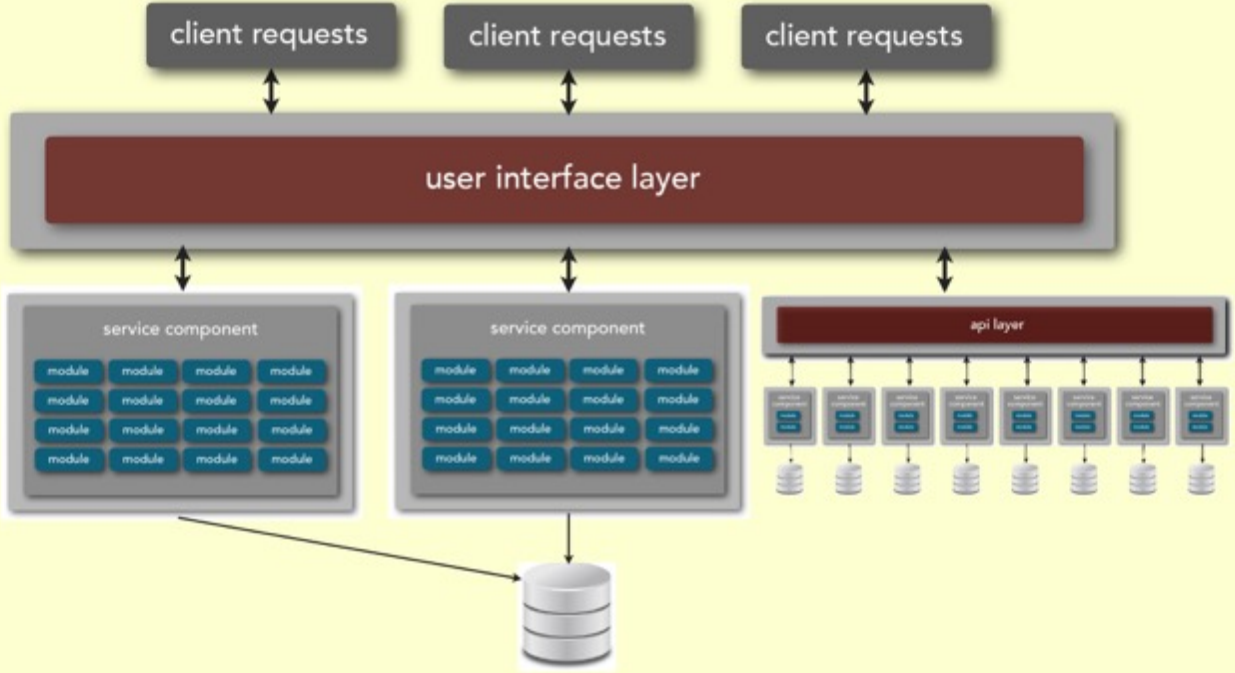


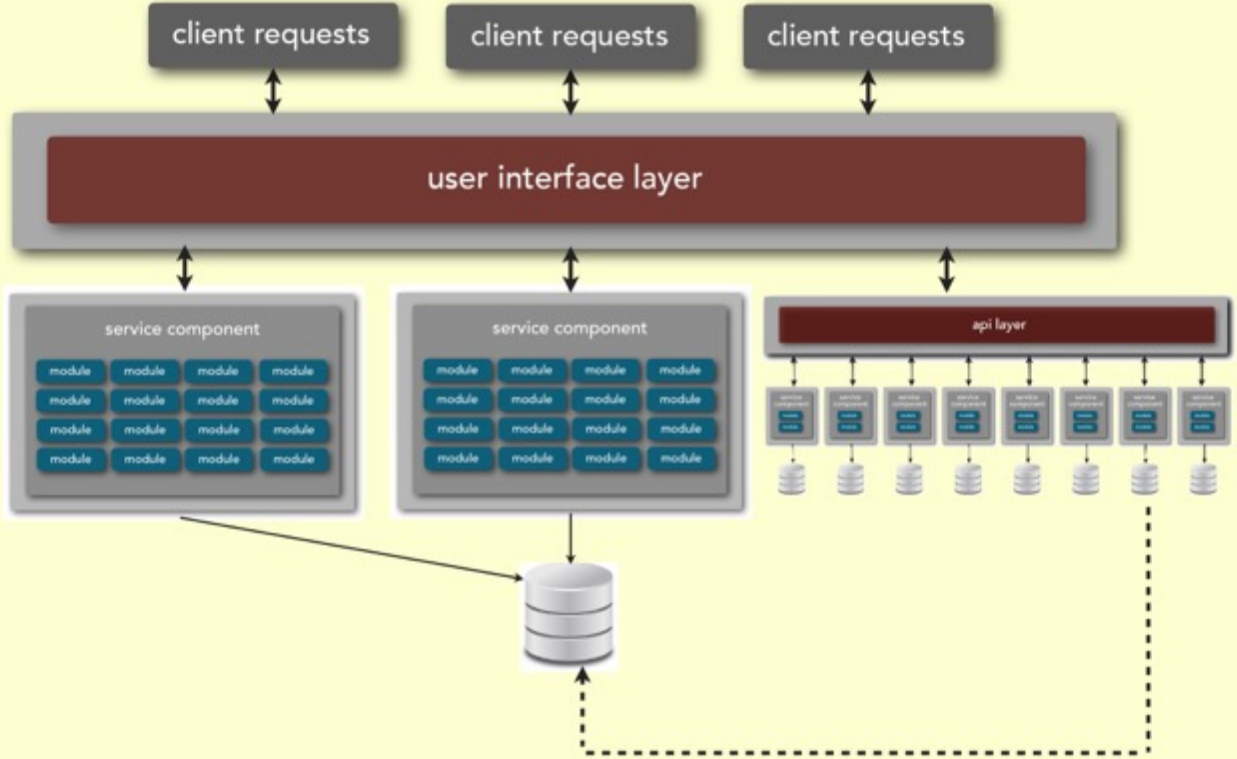
service-based architecture

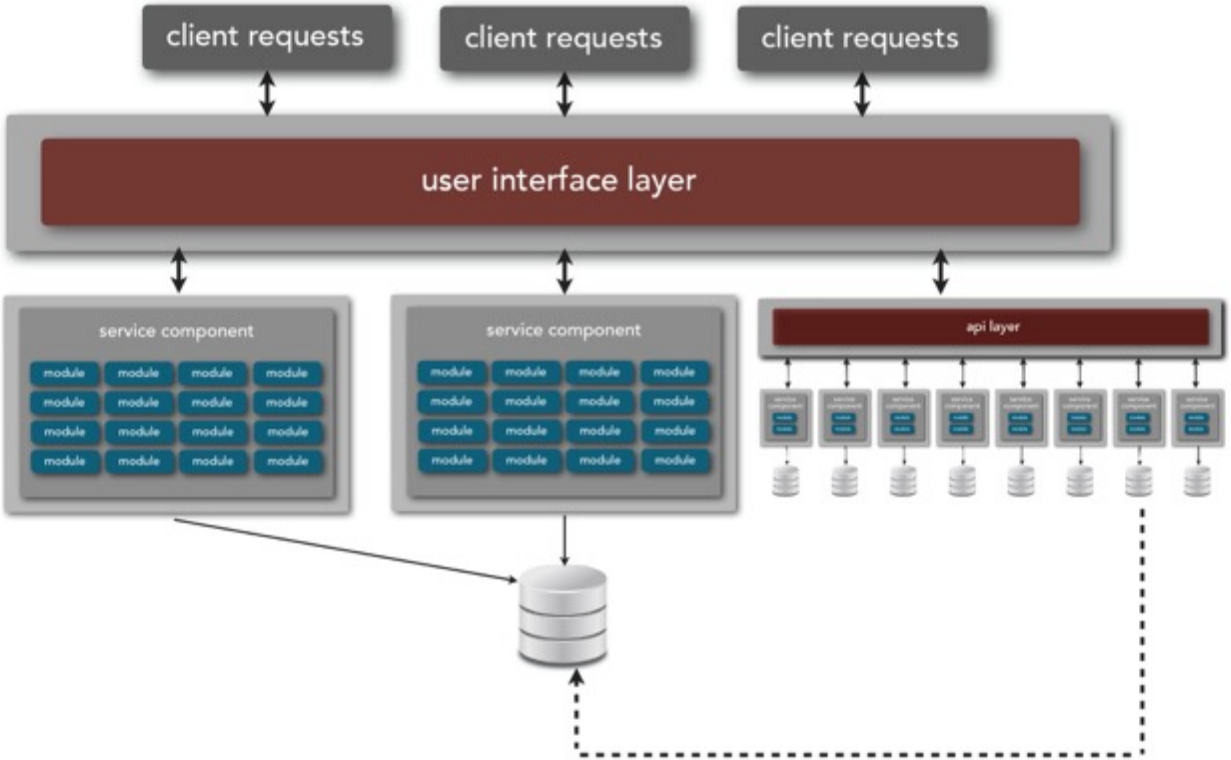

kiosk
public ui
receiving ui
recycling ui
quote service
item status service
receiving service
assessment service
recycling service
accounting service
reporting service
module


client requests
client requests
client requests
user interface layer
service component
module
service component
module
service component
module

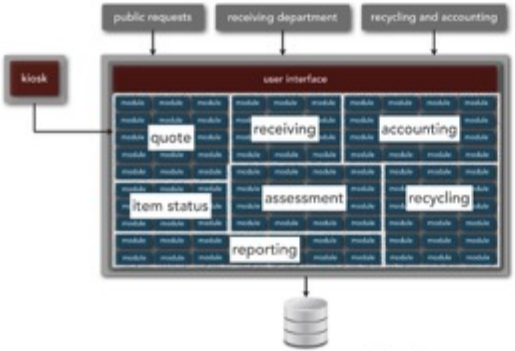

client requests
client requests
client requests
user interface layer
service component
module
service component
module
api layer


client requests
client requests
client requests
user interface layer
service component
module
service component
module
api layer


client requests
client requests
client requests
user interface layer
service component
module
service component
module
api layer

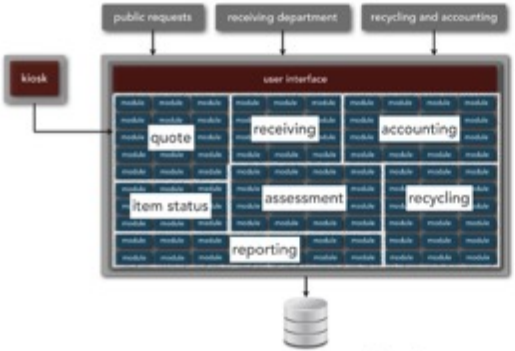

client requests
client requests
client requests
user interface layer
service component
module
service component
module
api layer

When Migrating Architecture...

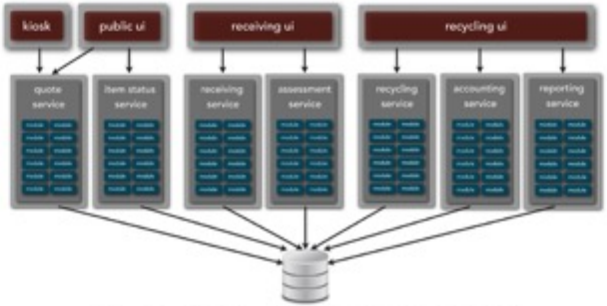


1. improve modularity

When Migrating Architecture...

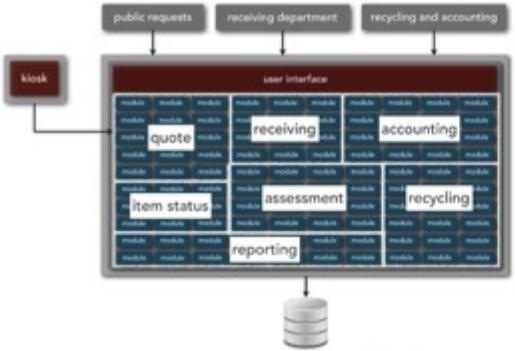


1. improve modularity

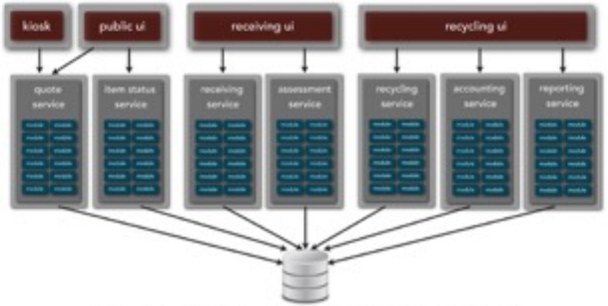


2. create service-based architecture

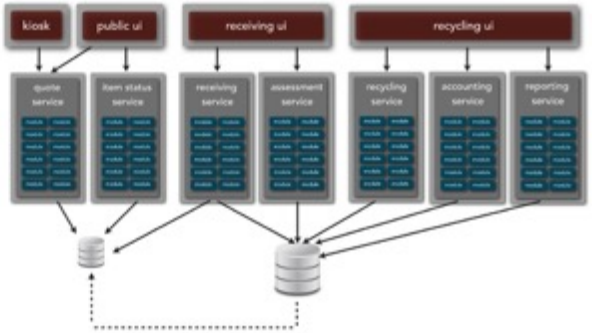
When Migrating Architecture...



1. improve modularity

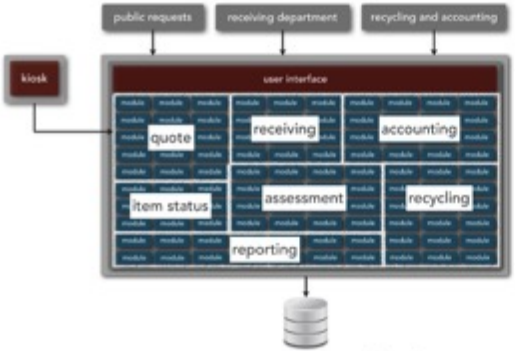


2. create service-based architecture

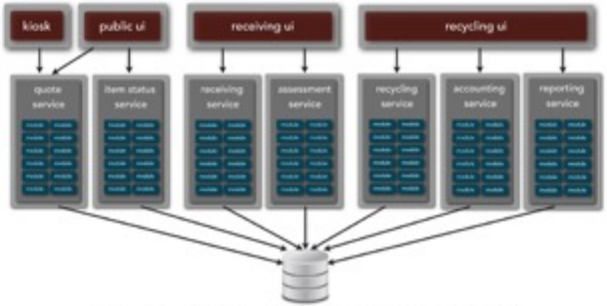


3. isolate quanta

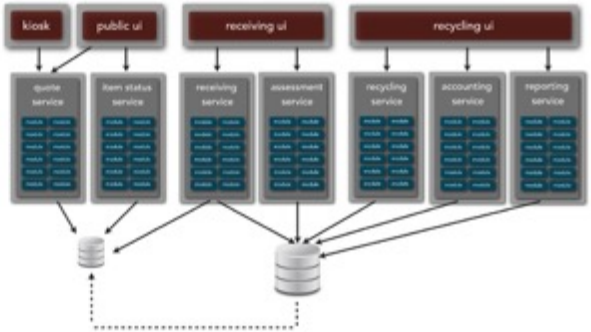
When Migrating Architecture...



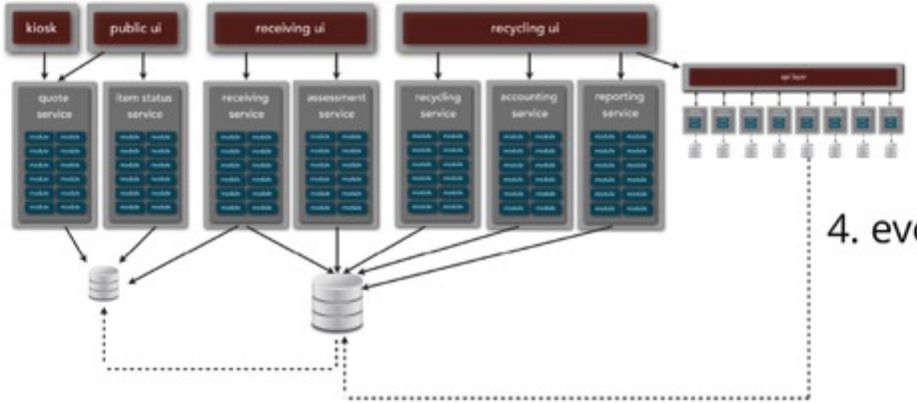
1. improve modularity



2. create service-based architecture



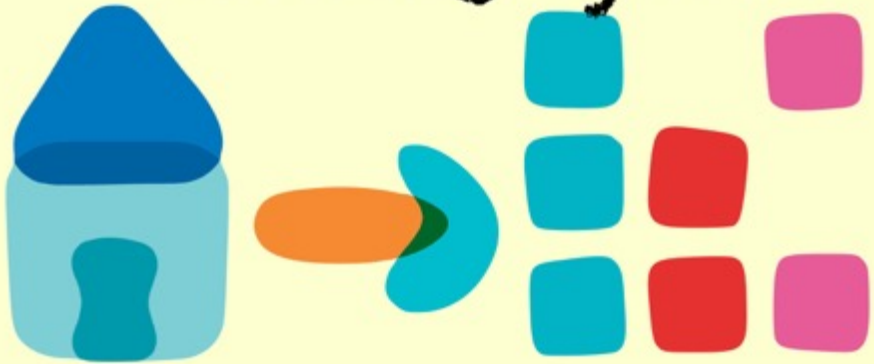
3. isolate quanta



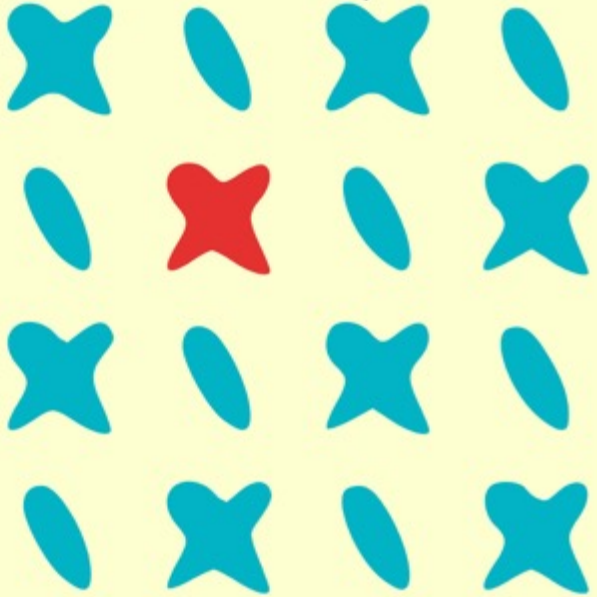
4. evolve via microservices

Penultimate

migrate critical existing systems

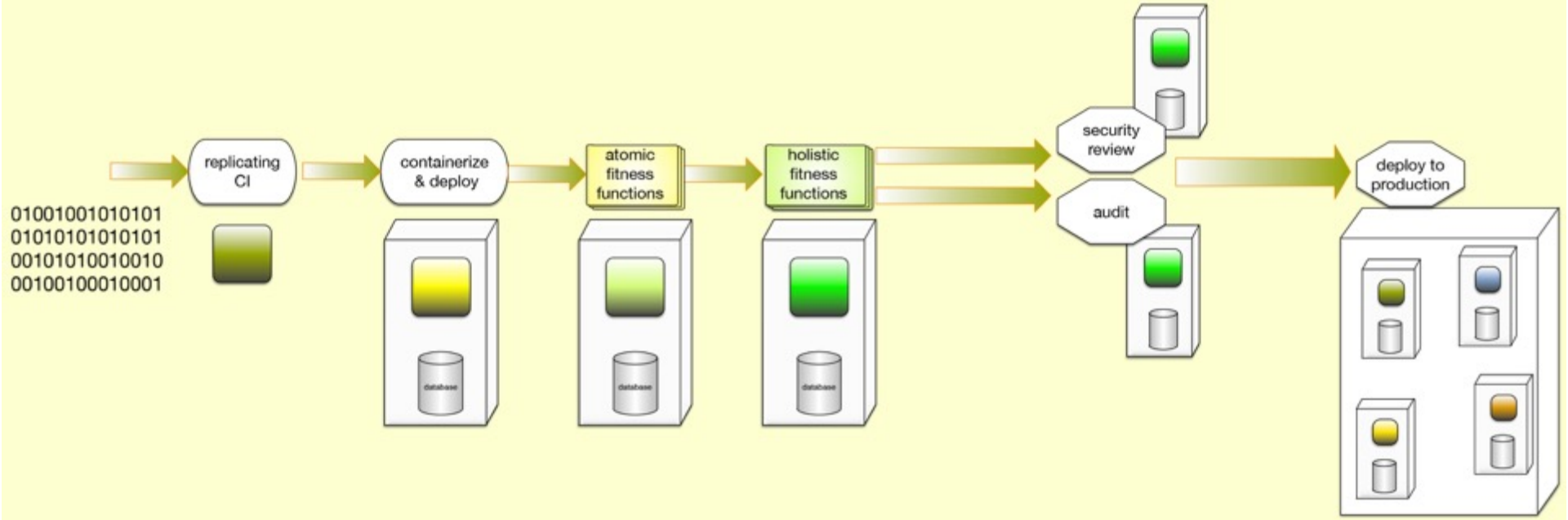


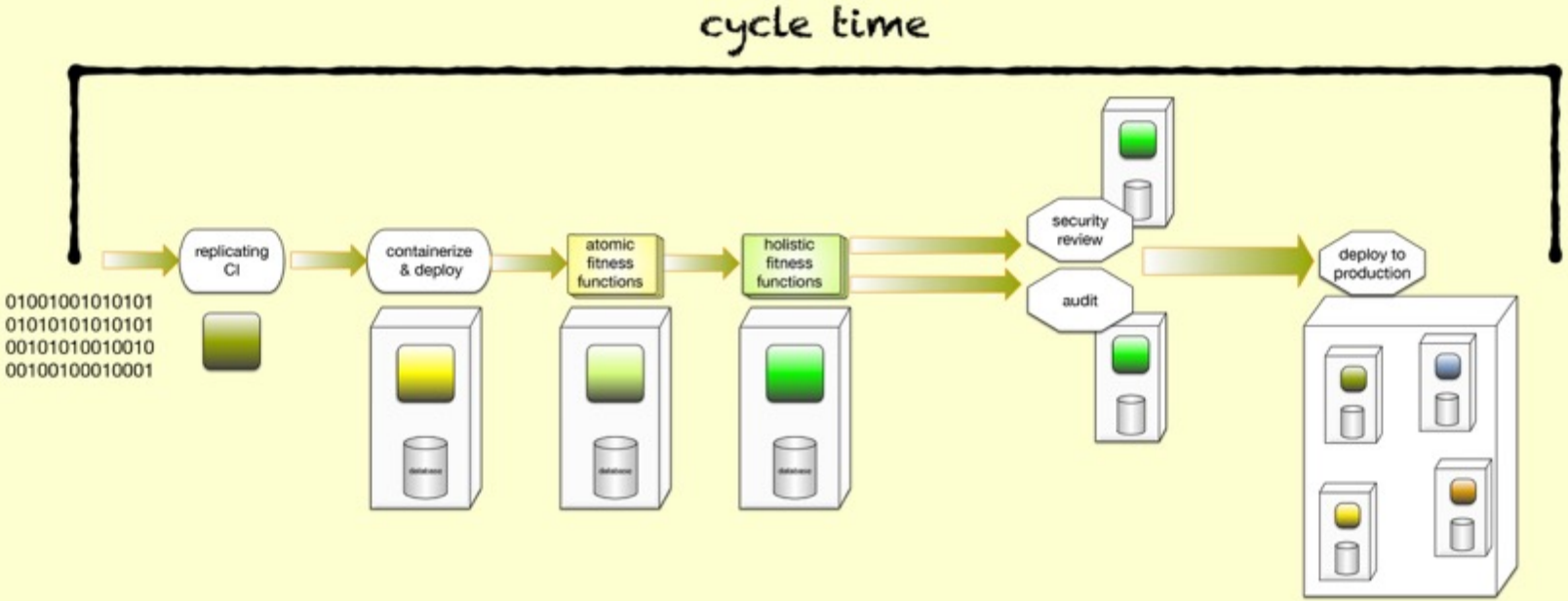
microservices for new development



Penultimate

Deployment Pipeline




cycle time
replicating CI
containerize & deploy
atomic fitness functions
holistic fitness functions
security review
audit
deploy to production
01001001010101
01010101010101
00101010010010
00100100010001

fitness function:

CycleTimeGuard

PenultimateWidgets has entered an extremely competitive market with a startup running popular flash promotions. The marketing department committed to keeping up with PenultimateWidgets own aggressive marketing campaign, which requires development teams to be able to deliver changes to the promotion pages within an hour. Thus, they want a guarantee that changes can always take place within an hour.

fitness function:

CycleTimeGuard

- utilize the good modularity of the monolith to build a deployment pipeline that automatically incorporates CMS changes
- continuously deploy new content for marketing pages as long as they pass smoke test
- measure cycle time from deployment pipeline for deployments for marketing sites
- save cycle time results to aggregate a weekly chart

Incidentally Coupled Teams

user interface

server-side

DBA

Conway's Law

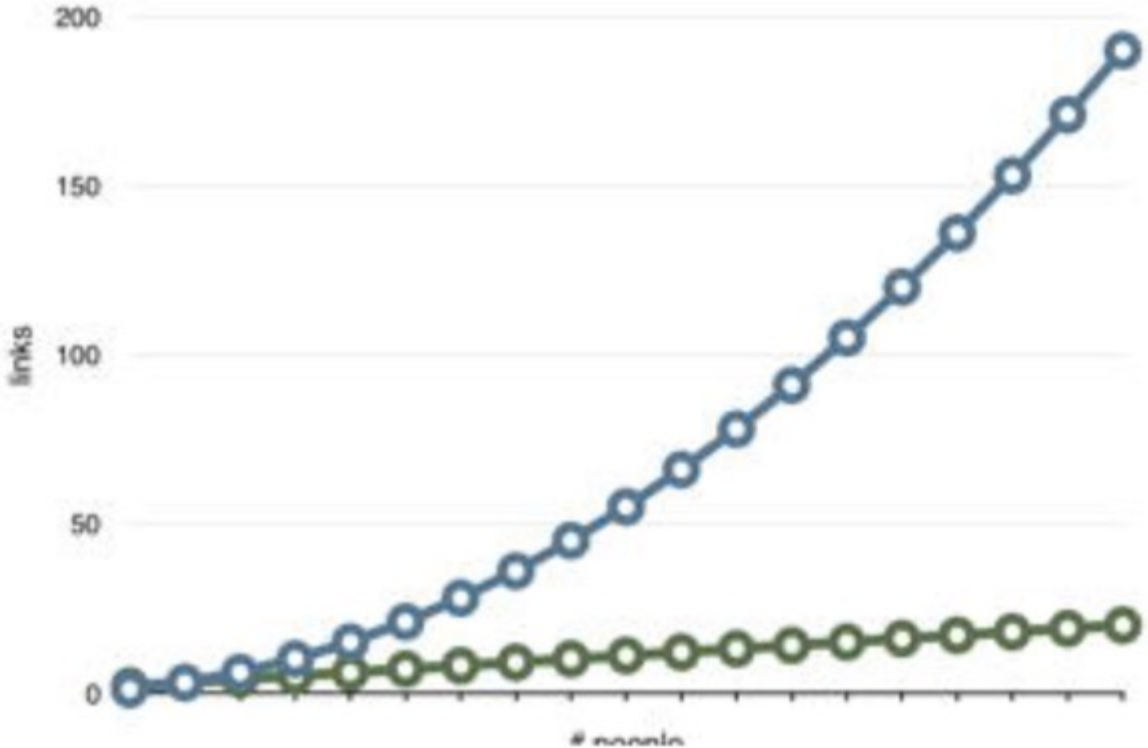
> *"organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations"*

Melvin Conway, 1968

en.wikipedia.org/wiki/Conway%27s_law

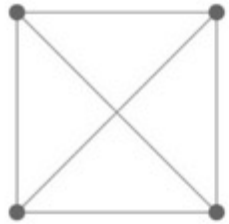
Low Efferent Coupling between Teams

$$\frac{n(n-1)}{2}$$

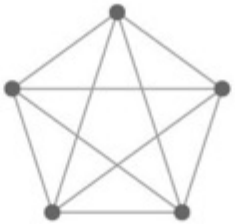


Low Efferent Coupling between Teams

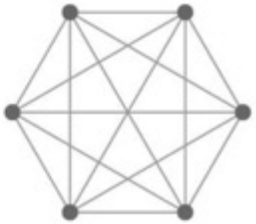
3 people, 3 lines



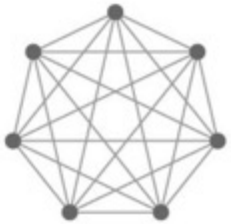
4 people, 6 lines



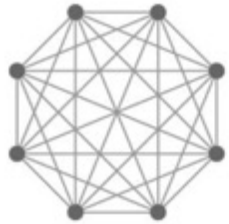
5 people, 10 lines



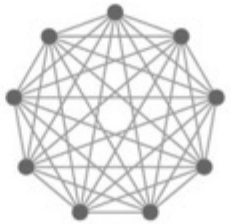
6 people, 15 lines



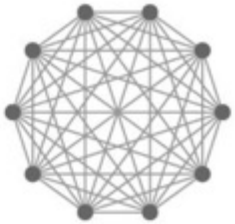
7 people, 21 lines



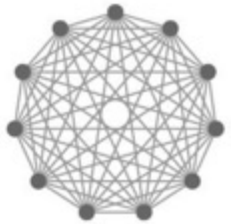
8 people, 28 lines



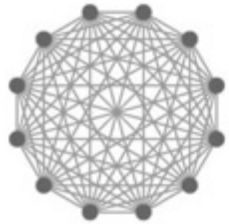
9 people, 36 lines



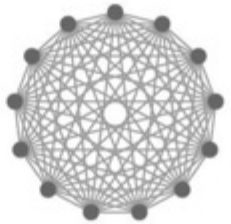
10 people, 45 lines



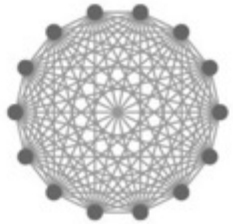
11 people, 55 lines



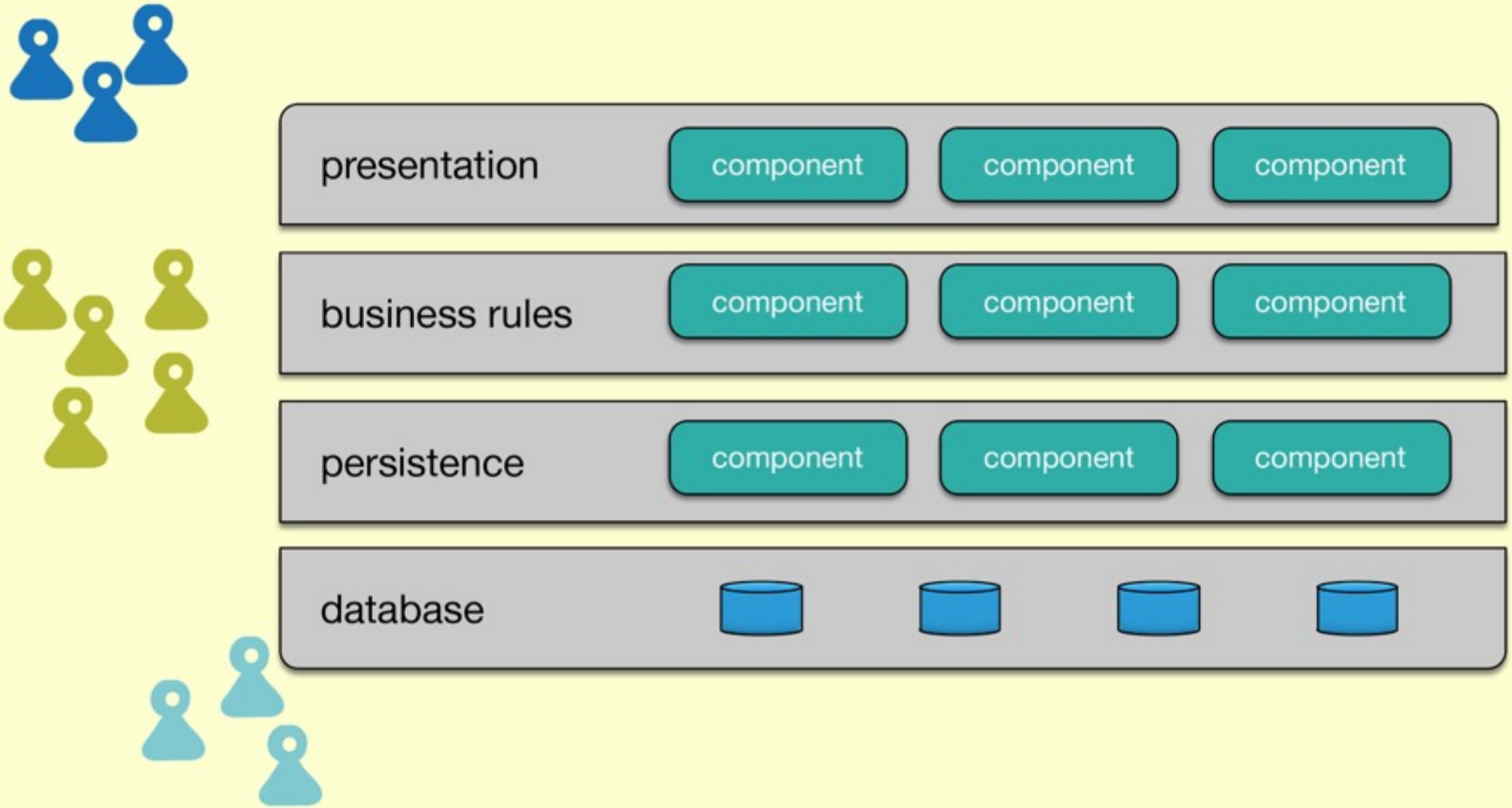
12 people, 66 lines

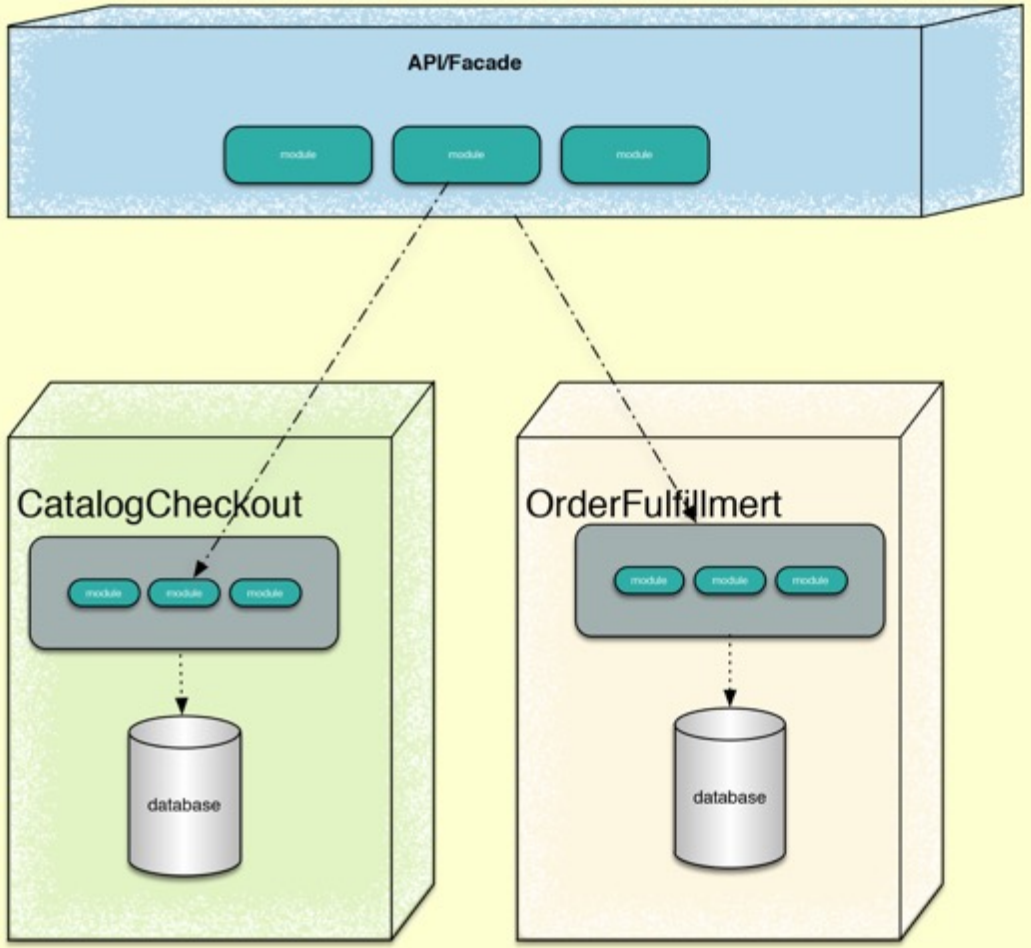


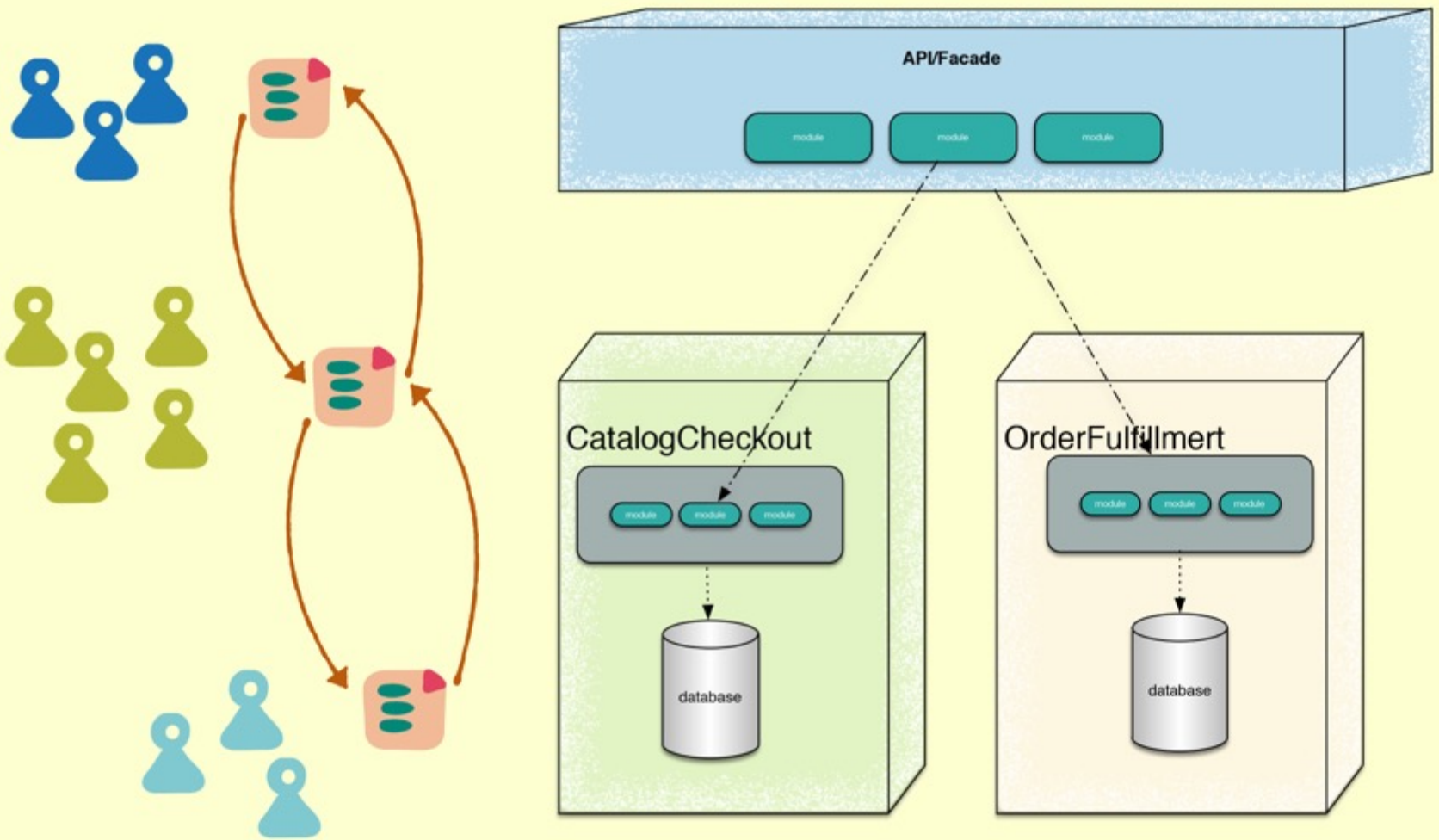
13 people, 78 lines

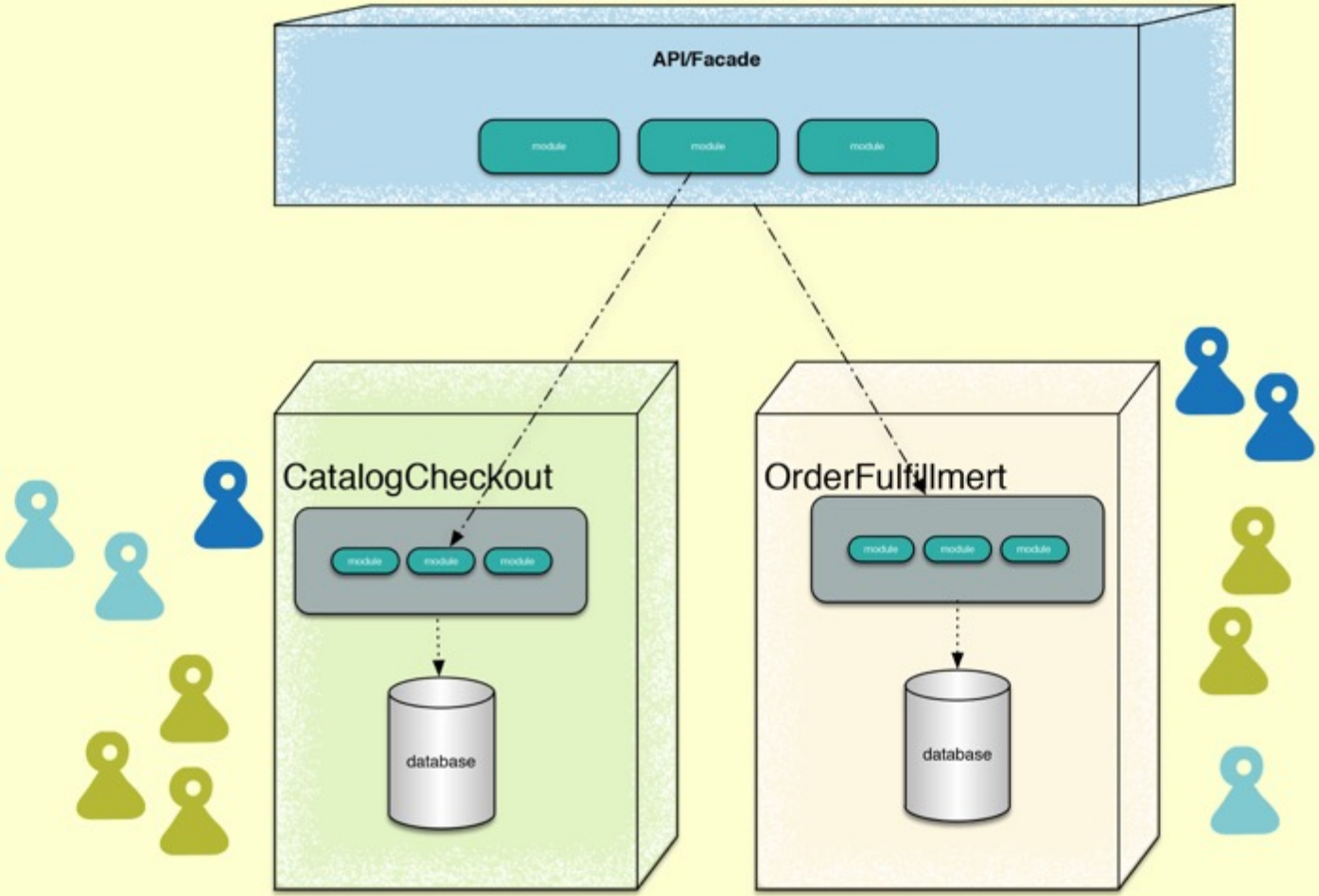


14 people, 91 lines

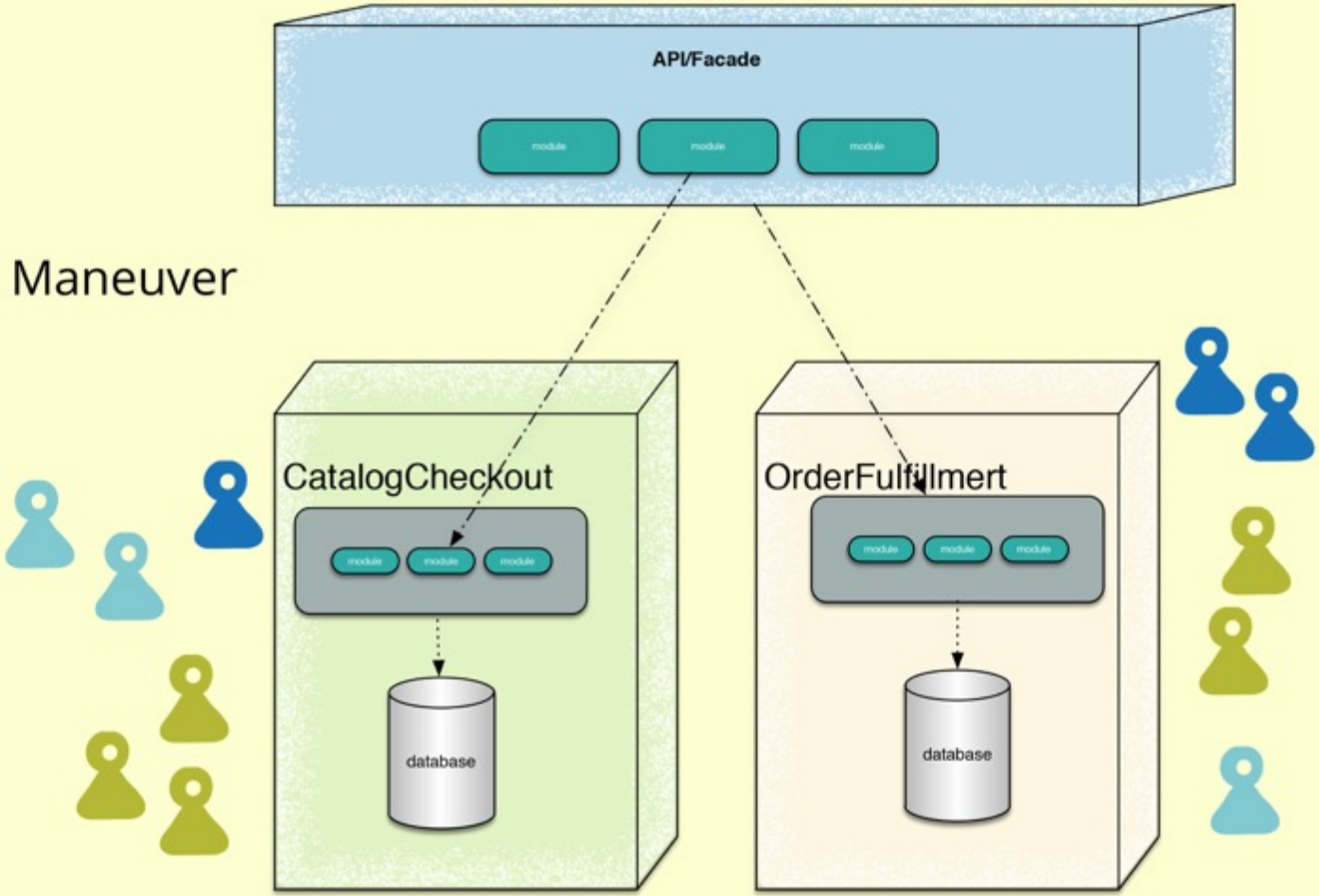

presentation
component
component
component
business rules
component
component
component
persistence
component
component
component
database

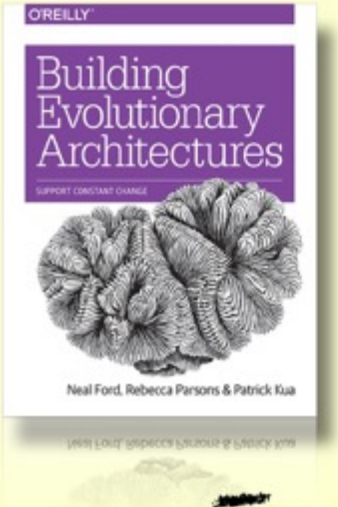

API/Facade
module
module
module
CatalogCheckout
module
module
module
database
OrderFulfillmert
module
module
module
database


API/Facade
module
module
module
CatalogCheckout
module
module
module
database
OrderFulfillmert
module
module
module
database


API/Facade
module
module
module
CatalogCheckout
module
module
module
database
OrderFulfillmert
module
module
module
database

Inverse Conway Maneuver





Experimentation is the only way to discover if something works within a particular organization.

Retrofitting Governance

Penultimate

fitness function:

Configure Some of the Things All of the Time

PenultimateWidgets has been burned in the past by having hard-coded constants in several places in their code base, governing both business cases (thresholds, maximums, etc.) and operational concerns. Architects need a way to prevent developers from accidentally hard-coding critical values.

fitness function:

Configure Some of the Things All of the Time

- Upon check-in:
  - scan the list of dynamic values in the code base and see if any of them are assigned to (or their setter is called) or a new instance is instantiated via a constructor that provides values
  - ensure that all assignments for dynamic values are assigned via the API that handles configuration, not with hard-coded values
  - scan the definitions of infrastructure to look for dynamic values and ensure they are assigned properly before building the environment
  - scan the configuration file and ensure that all defined dynamic values are present and have values

fitness function:

Audit the Accounting

PenultimateWidgets has been a Bad Company (cheating on some complex accounting rules), but they have reformed their ways and paid their penalties. Now, however, as part of the settlement, they agree to put auditing measures in place around code that handles accounting rules. State auditors come once a quarter to look everything over, but PenultimateWidgets has found it too much of a chore to delay preparation until near audit time; they want a more automated way to verify things.

fitness function:

Audit the Accounting

- During check-in, look at git logs to see if any files in the affected packages were modified
- for any modified file, add it to a database of files to check by the auditors
- the tech lead code reviews changes for the last week before standup
- at audit time, show each of the changed files to auditors and allow them access to the repository to verify that no other files were changed

fitness function:

Debug All the Things

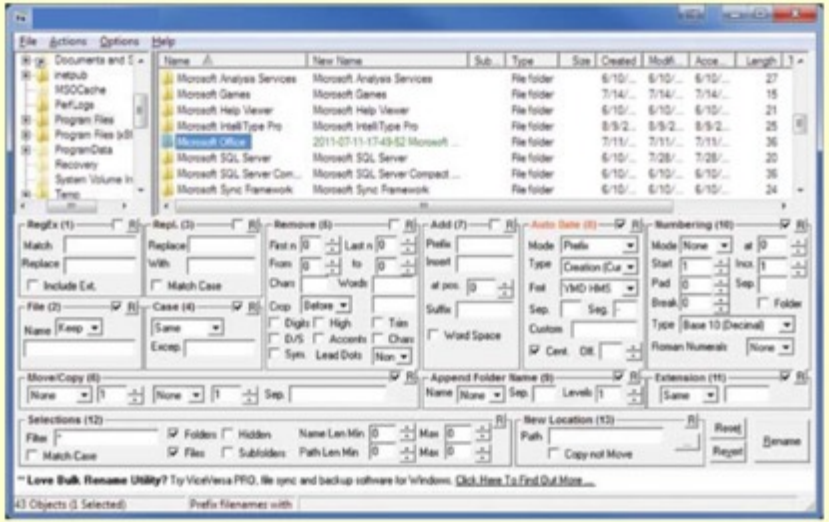
PenultimateWidgets wants to ensure that both Java and Javascript applications support remote debugging in the development environment to aid in faster customer support resolution.

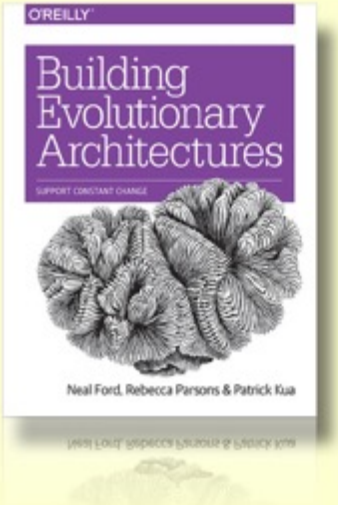
fitness function:

Debug All the Things

- for Java applications, create a monitor that runs in development and QA that checks to ensure the remote debugging ports are available and listening for all deployed Java applications
- create a symmetrical monitor for production systems to ensure the remote debugging ports are NOT enabled in production
- For Javascript, create a special flag variable declaration at the top of the file that is not used anywhere else (meaning the minimizer will remove it). For development and QA environments, check to make sure the special flag variable is present
- include source maps for obfuscated code to map problems back to the source

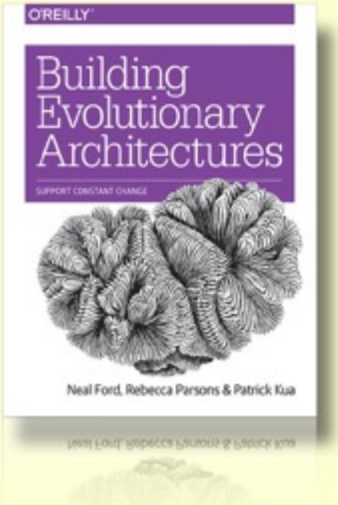
What to Port?


File Actions Options Help
Cash Flow
Reports
Companies
Settings
Users
Account
Add Income
Add Expense
Export
INCOME
EXPENSES
CONSULTING
PRODUCTS
SERVICES
PAYROLL
RENT



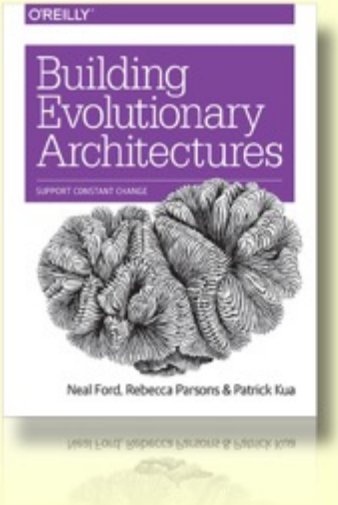
Experimentation!

- add logging to the existing application to aggregate what menu items users choose and the user group



Experimentation!

- add logging to the existing application to aggregate what menu items users choose and the user group
- publish the log to a known URL at PenultimateWidgets



Experimentation!

- add logging to the existing application to aggregate what menu items users choose and the user group
- publish the log to a known URL at PenultimateWidgets
- aggregate the logs to get usage and user coverage

fitness function:

Make Sure New Really Replaces Old

PenultimateWidgets legacy accounting system grew over many years, with arcane business rules scattered in many places—no one really understands how everything works anymore. They worked for several years replacing it, but worry that the new system cannot replicate the byzantine business rules that their clients have come to depend upon.

fitness function:

Make Sure New Really Replaces Old

- For each discrete function covered under this rule, create a functional test that treats each system (old and new) as a black box, looking only at the results of each.
- Provision the old system (perhaps using the Legacy in the Box pattern) and new in an integration testing environment and run the functional replacement tests.
- Stories are not marked complete until the functional tests run.

fitness function:

Replace the Crufty Core

PenultimateWidgets built a network scanning tool many, many years ago to look for suspicious activity at the network packet level. To achieve the proper performance, the code was written and has been maintained in increasingly complex C code. PenultimateWidgets needs to replace the code but is nervous about introducing bugs and/or harming performance or throughput.

fitness function:

Replace the Crufty Core

- Built fitness functions around the existing C code to ensure performance and throughput values.
- Build a spike solution in both Java and Go (and perhaps others) and run them against the fitness functions to see if they can meet the threshold
- Use Scientist (https://github.com/github/scientist) port Scientist4J (https://github.com/rawls238/Scientist4J) to verify the new code against the old code in production

While We’re at it…

fitness function:

A11y All the Things

PenultimateWidgets has decided that all new development of user interfaces must support accessibility, both web and desktop.

fitness function:

A11y All the Things

- add a fitness function to the deployment pipeline for web projects that runs pa11y (http://pa11y.org/) to ensure compliance
- for Java, use PMD to check code within the user interface and look for control instantiation
- check to make sure every control has the AccessibleName property set

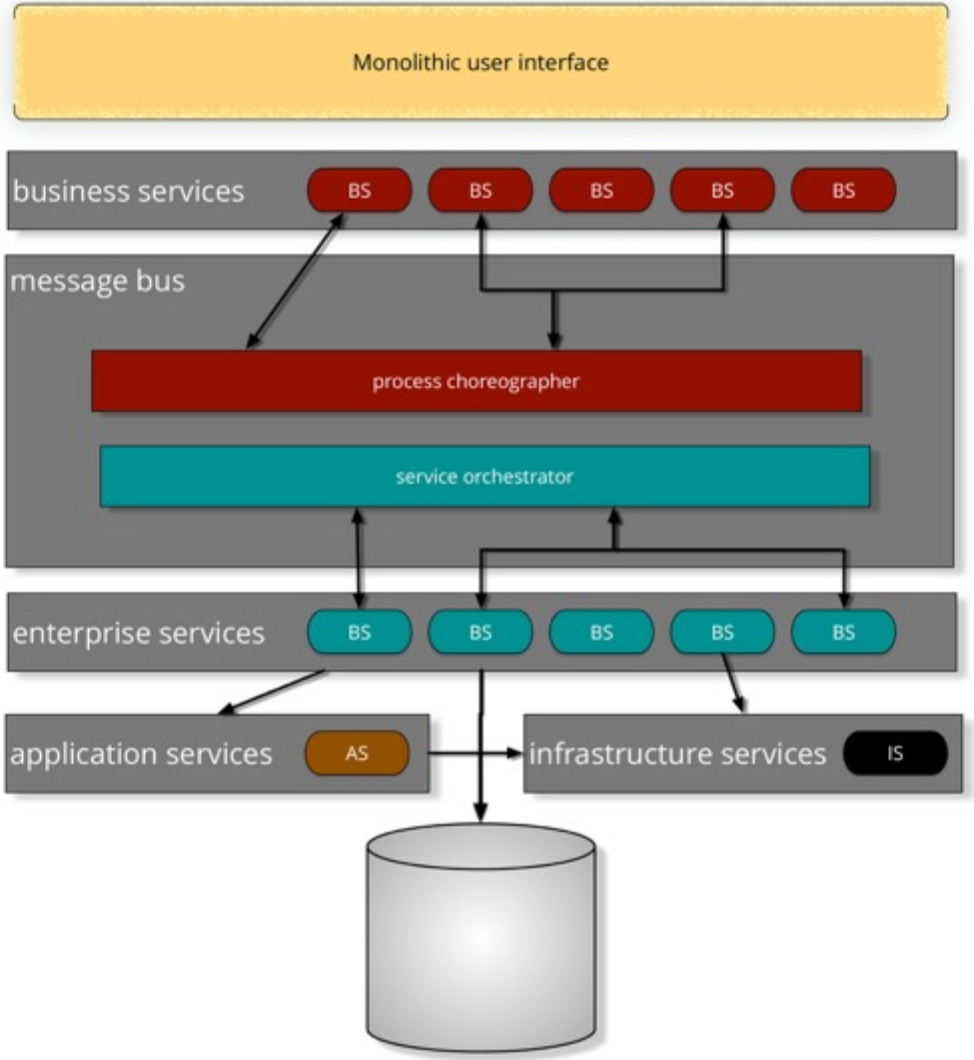
Mitigating the Unknown

reuse

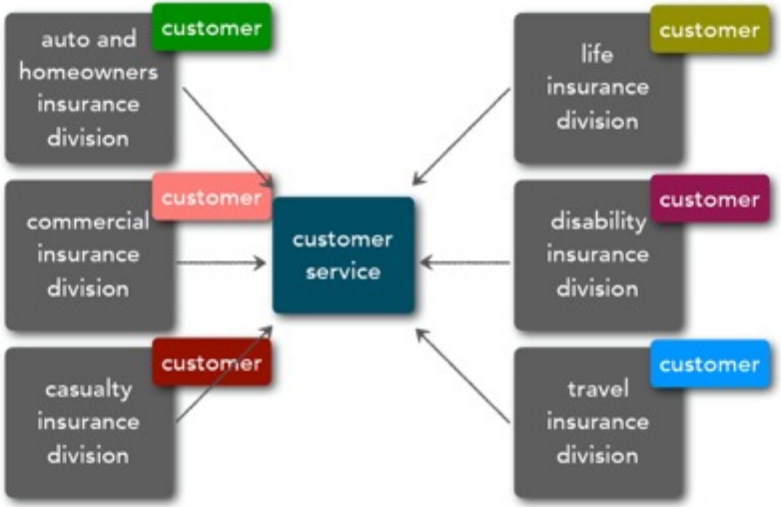
Software reuse is more like an organ transplant than snapping together Lego blocks.

John D. Cook

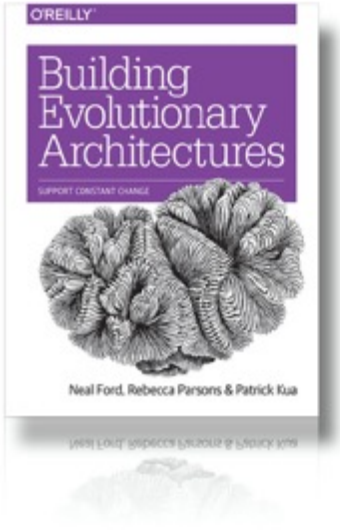
http://www.johndcook.com/blog/2011/02/03/lego-blocks-and-organ-transplants/


Monolithic user interface
business services
BS
BS
BS
BS
BS
message bus
process choreographer
service orchestrator
enterprise services
BS
BS
BS
BS
BS
application services
AS
infrastructure services
IS

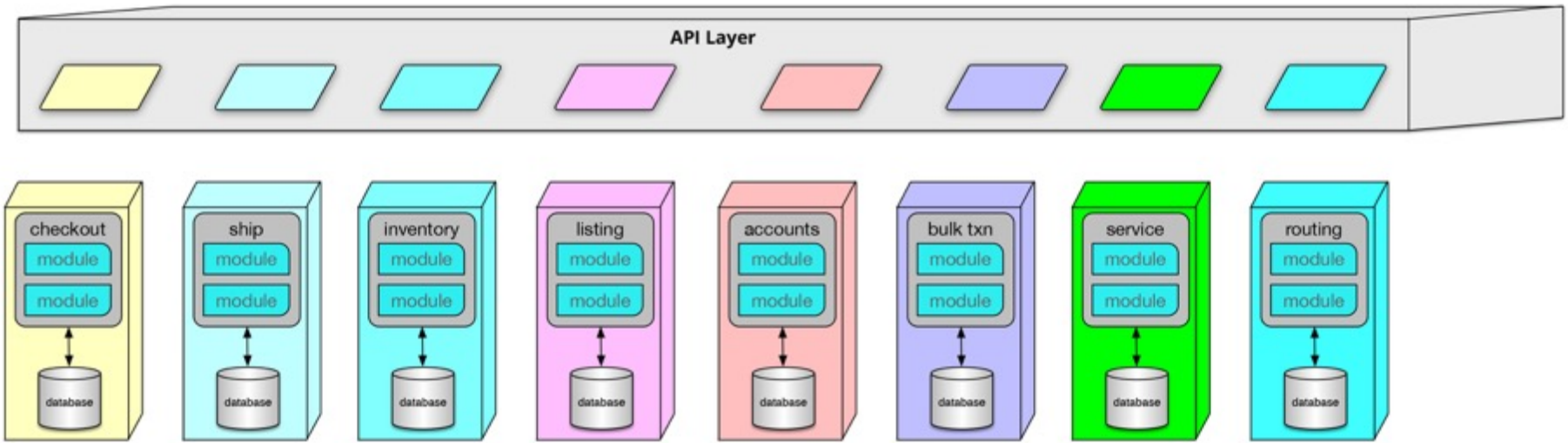

auto and homeowners insurance division
commercial insurance division
casualty insurance division
customer service
life insurance division
disability insurance division
travel insurance division


auto and homeowners insurance division
customer
commercial insurance division
customer
casualty insurance division
customer
customer service
life insurance division
customer
disability insurance division
customer
travel insurance division
customer

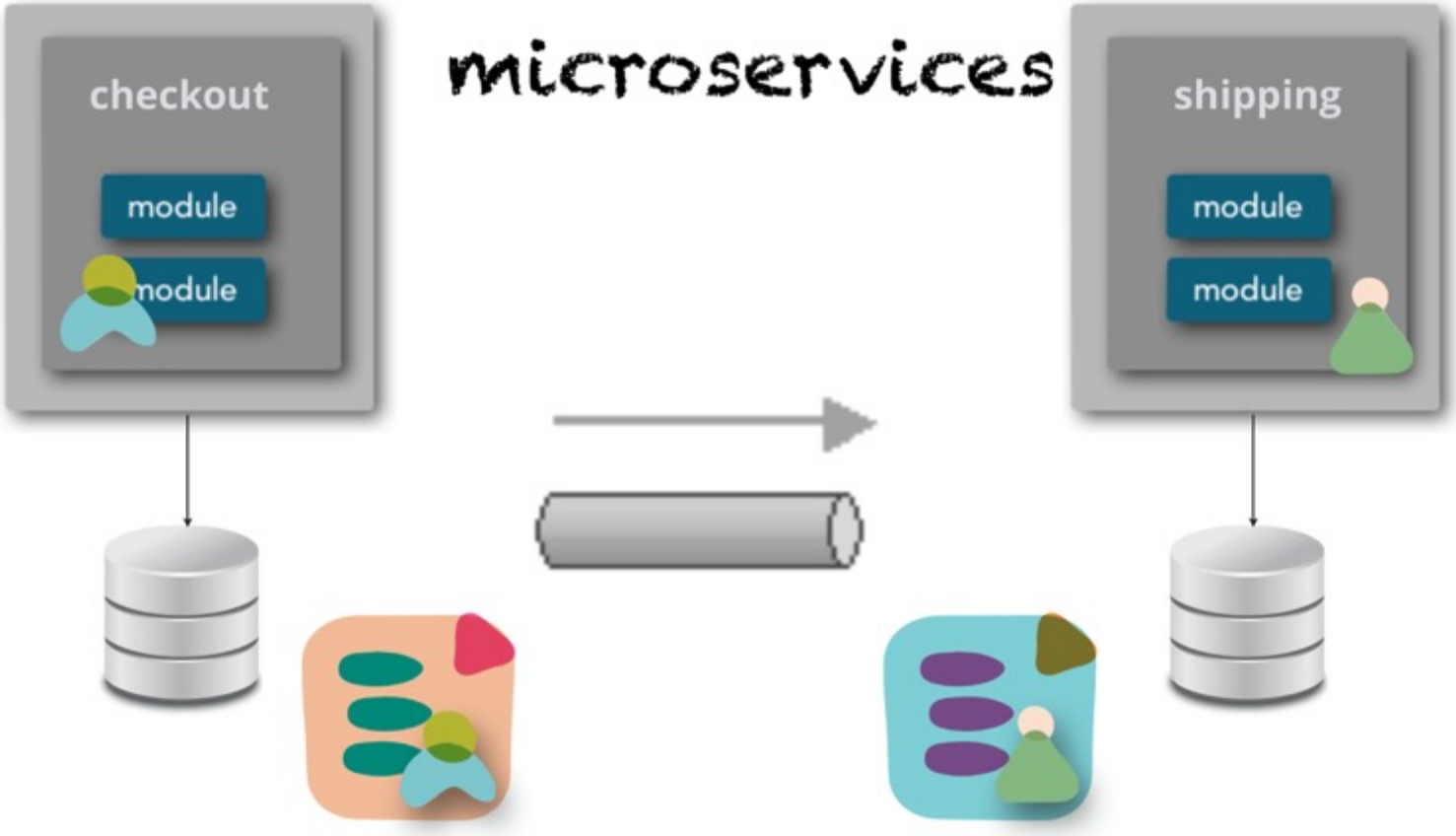
Code Reuse (Over Time)

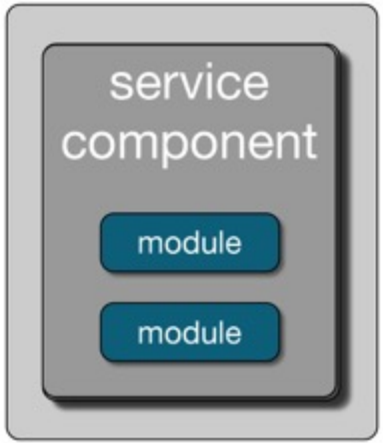


The more reusable code is, the less usable it is.


API Layer
checkout
module
module
database
ship
module
module
database
inventory
module
module
database
listing
module
module
database
accounts
module
module
database
bulk txn
module
module
database
service
module
module
database
routing
module
module
database

Code Reuse (Over Time)




service
component
module
module

Sidecars (Service Templates)

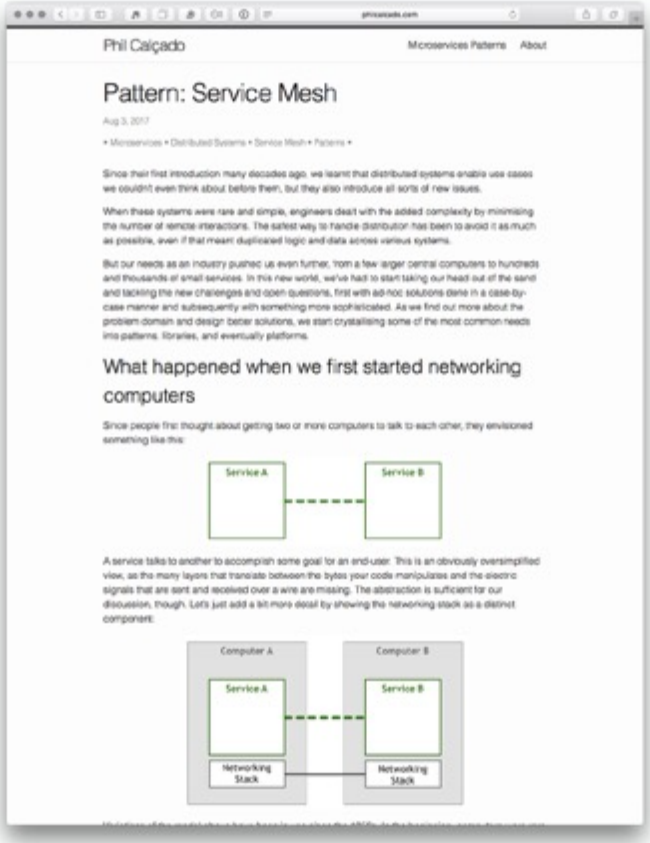
Sidecars (Service Templates)

Sidecars (Service Templates)

https://projects.spring.io/spring-boot/

http://www.dropwizard.io/

the rise of service meshes



Phil Calçado

Microservices Patterns About

Pattern: Service Mesh

Aug 3, 2017

• Microservices • Distributed Systems • Service Mesh • Patterns •

Since their first introduction many decades ago, we learnt that distributed systems enable use cases we couldn't even think about before them, but they also introduce all sorts of new issues.

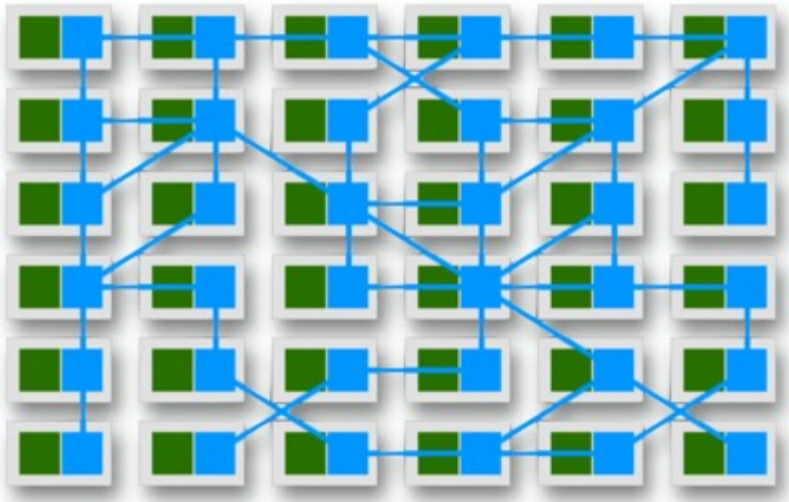
When these systems were rare and simple, engineers dealt with the added complexity by minimising the number of remote interactions. The safest way to handle distribution has been to avoid it as much as possible, even if that meant duplicated logic and data across various systems.

But our needs as an industry pushed us even further, from a few larger central computers to hundreds and thousands of small services. In this new world, we've had to start taking our head out of the sand and tackling the new challenges and open questions, first with ad-hoc solutions done in a case-by-case manner and subsequently with something more sophisticated. As we find out more about the problem domain and design better solutions, we start crystallising some of the most common needs into patterns, libraries, and eventually platforms.

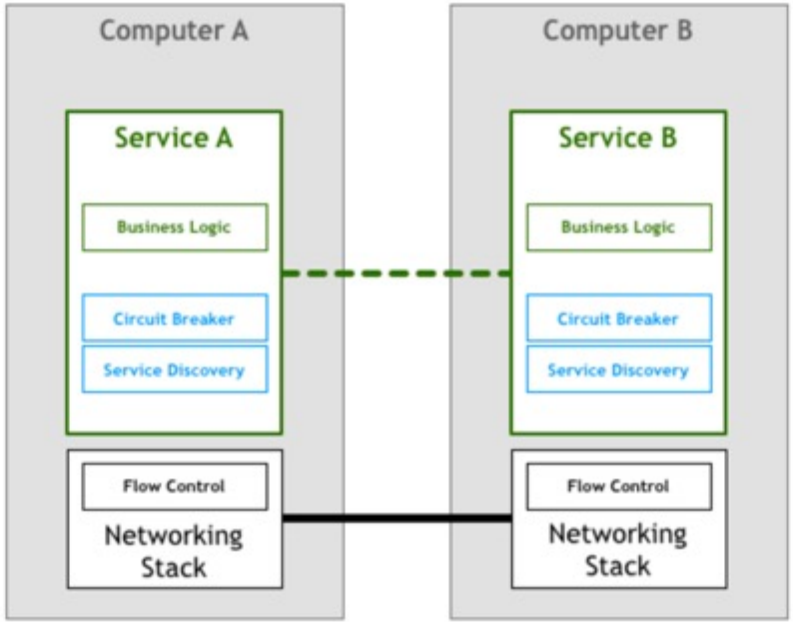
What happened when we first started networking computers

Since people first thought about getting two or more computers to talk to each other, they envisioned something like this:

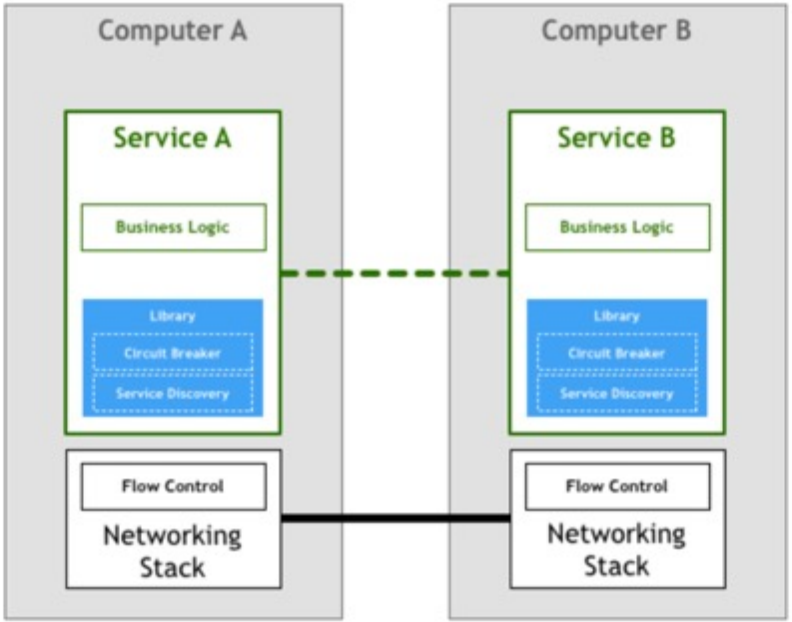
A service talks to another to accomplish some goal for an end-user. This is an obviously oversimplified view, as the many layers that translate between the bytes your code manipulates and the electric signals that are sent and received over a wire are missing. The abstraction is sufficient for our discussion, though. Let's just add a bit more detail by showing the networking stack as a distinct component:



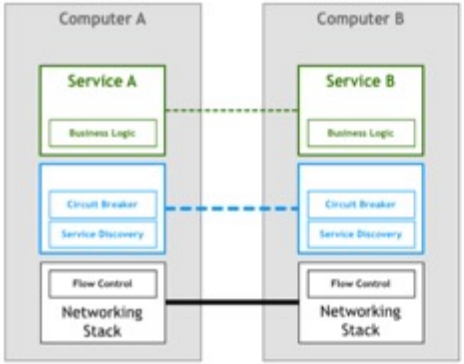
the rise of service meshes

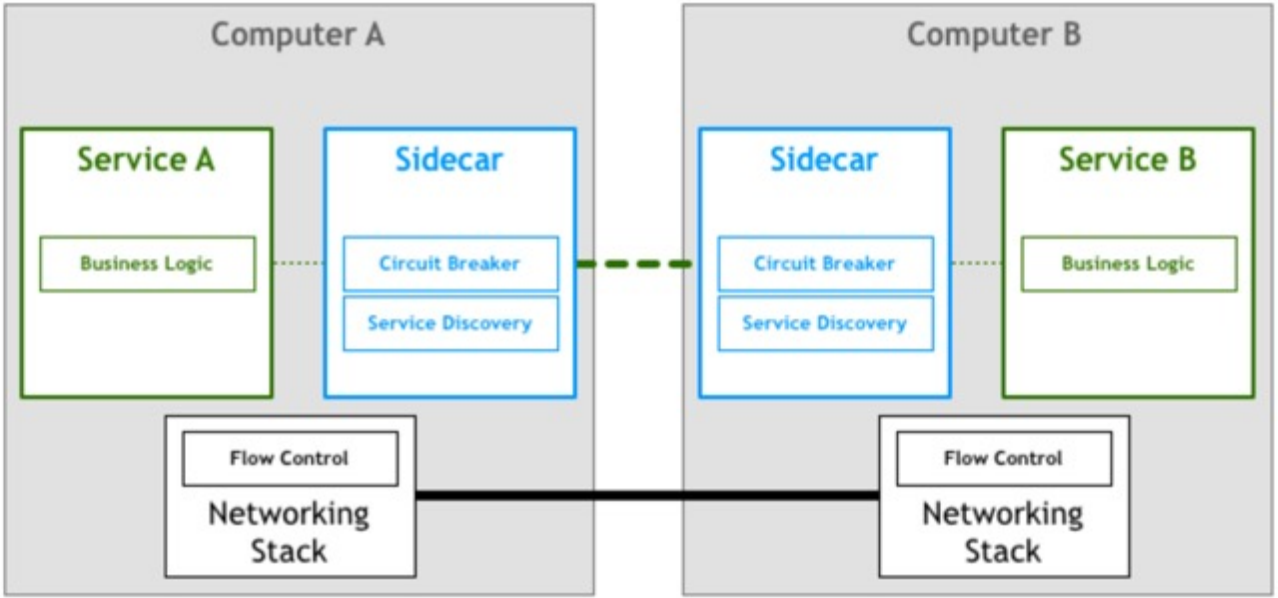


the rise of service meshes

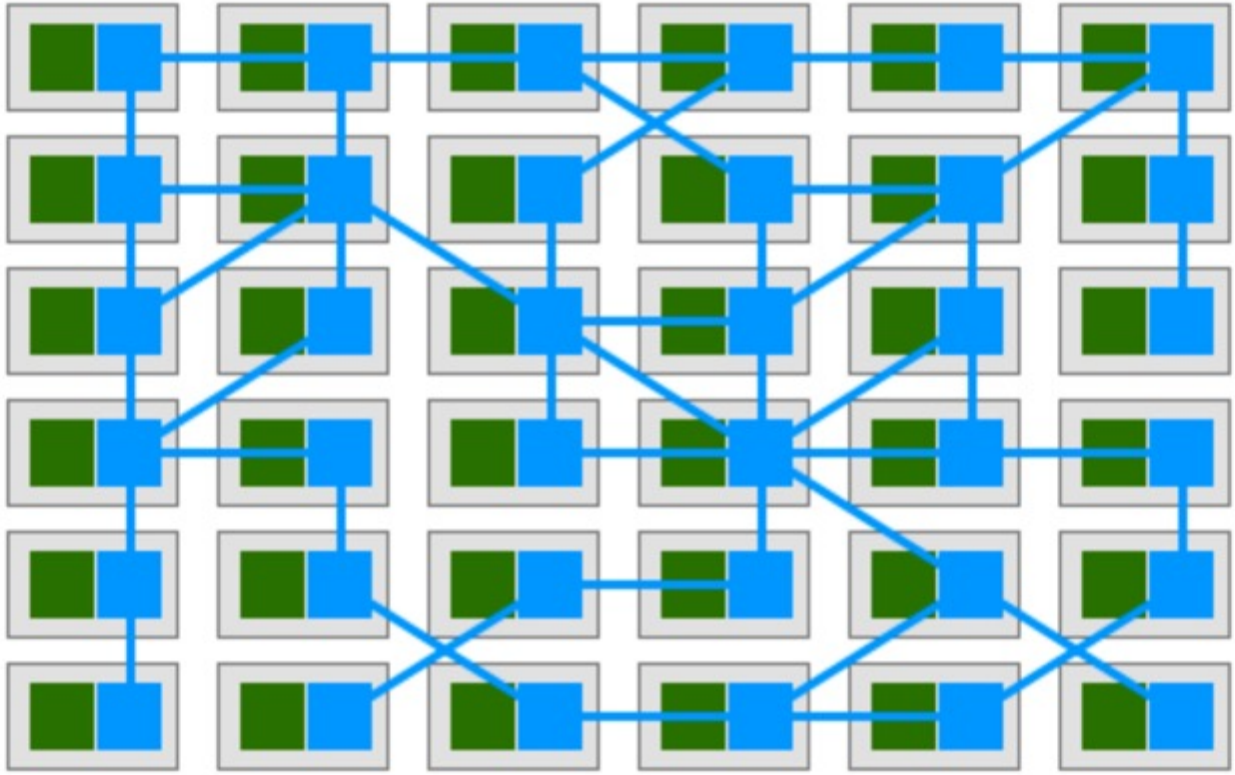


the rise of service meshes

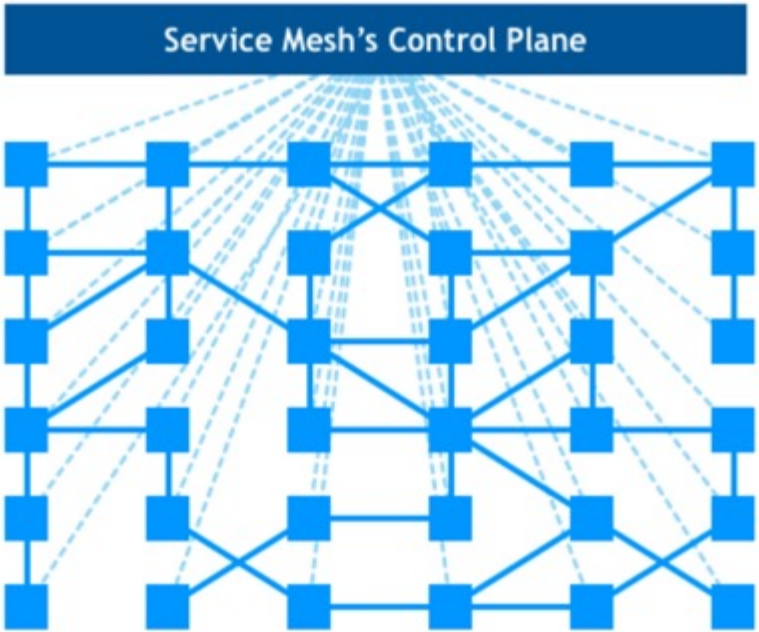


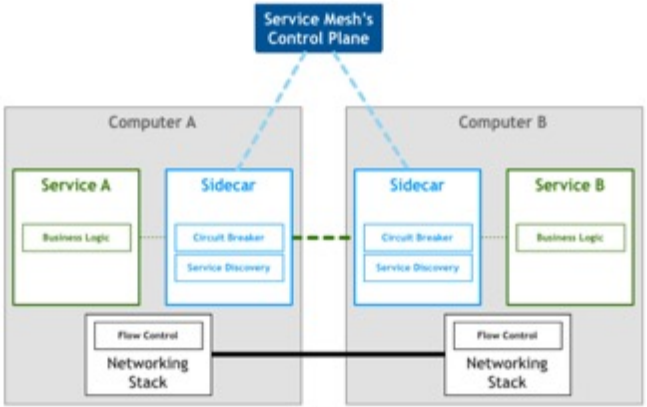


the rise of service meshes



the rise of service meshes





fitness function:

Elasticity of Microservices

PenultimateWidgets utilizes a microservices architecture and want to ensure that as new requests appear that the architecture elastically scales.

Penultimate fitness function:

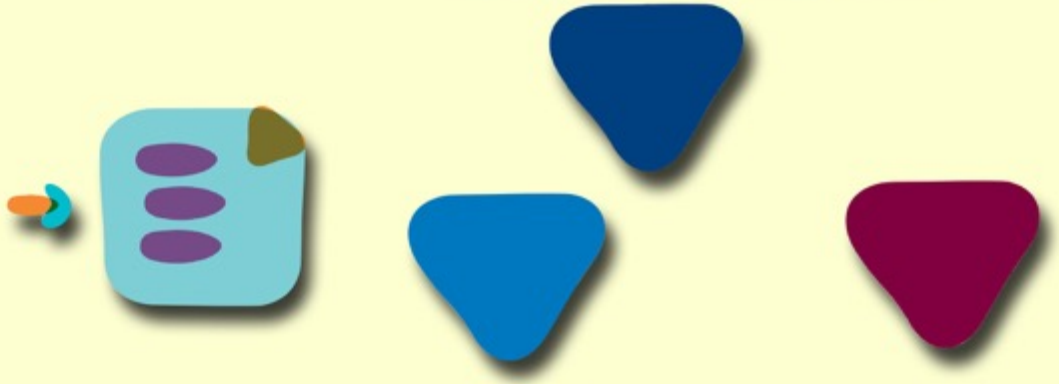
Elasticity of Microservices

PenultimateWidgets utilizes a microservices architecture and want to ensure that as new requests appear that the architecture elastically scales.

Penultimate fitness function:

Elasticity of Microservices

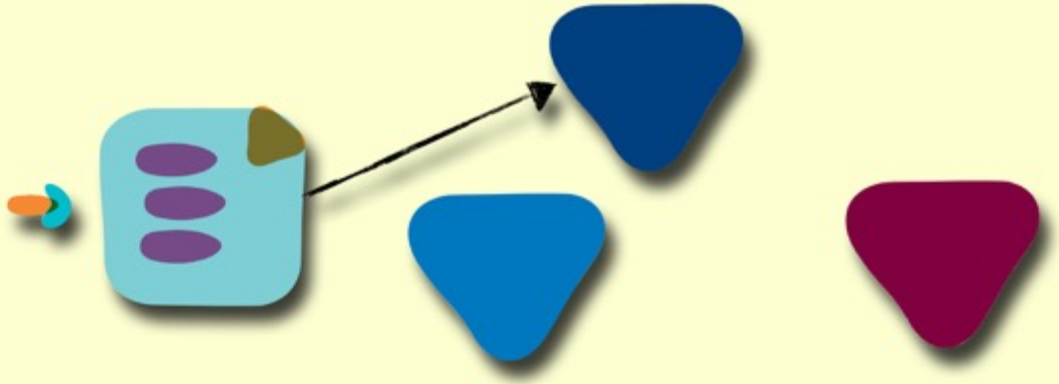
PenultimateWidgets utilizes a microservices architecture and want to ensure that as new requests appear that the architecture elastically scales.



Penultimate fitness function:

Elasticity of Microservices

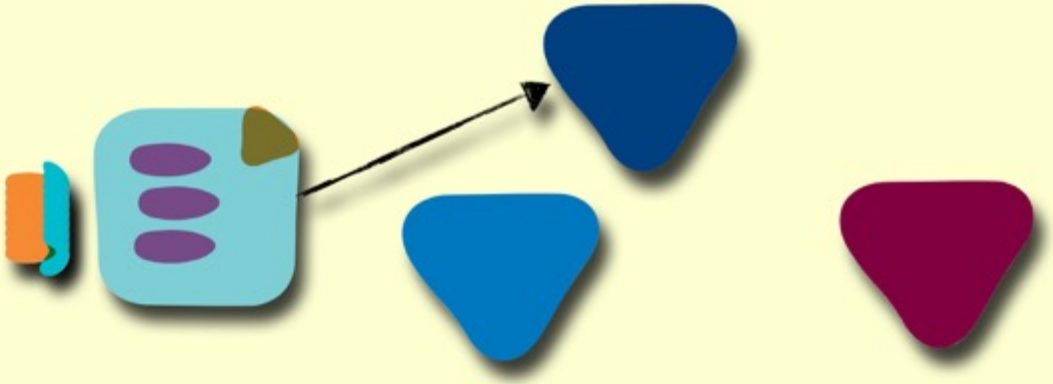
PenultimateWidgets utilizes a microservices architecture and want to ensure that as new requests appear that the architecture elastically scales.



Penultimate fitness function:

Elasticity of Microservices

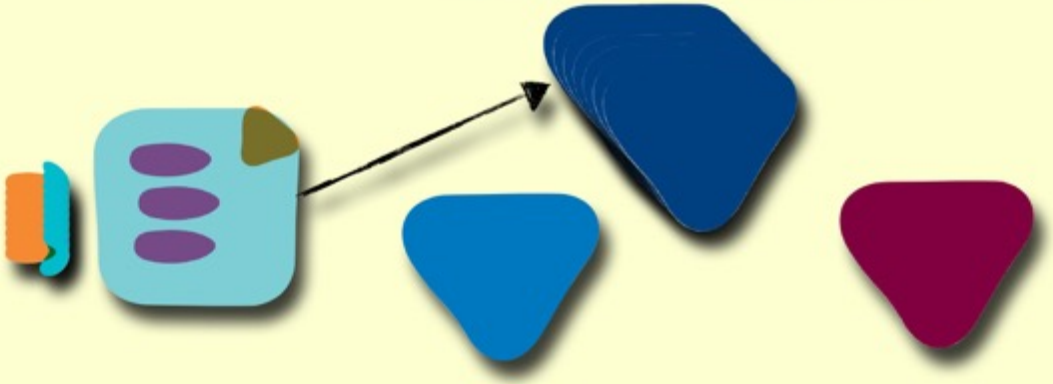
PenultimateWidgets utilizes a microservices architecture and want to ensure that as new requests appear that the architecture elastically scales.



Penultimate fitness function:

Elasticity of Microservices

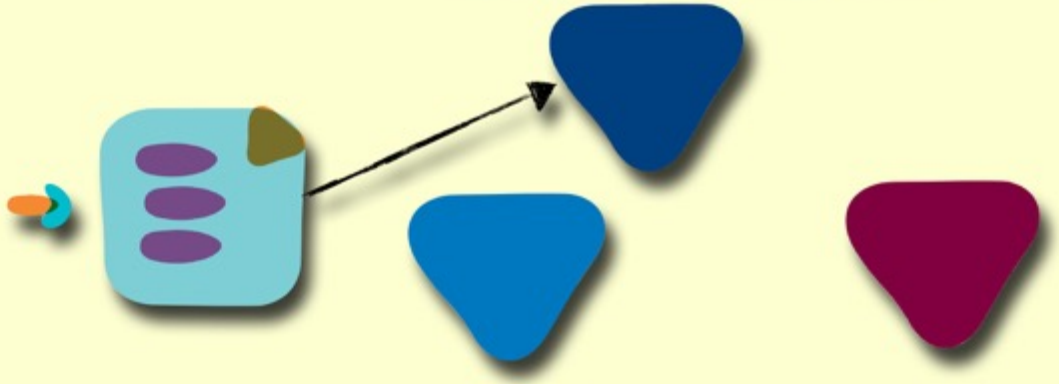
PenultimateWidgets utilizes a microservices architecture and want to ensure that as new requests appear that the architecture elastically scales.



Penultimate fitness function:

Elasticity of Microservices

PenultimateWidgets utilizes a microservices architecture and want to ensure that as new requests appear that the architecture elastically scales.



fitness function:

Elasticity of Microservices

- build a Hello, World service to allow instantiation of the service template
- deploy the service and direct enough requests to it to trigger elasticity in the naming service
- verify that the number of instances rose properly

fitness function:

Deterministic Distributability

PenultimateWidgets has been burned in the past by hard-coded strings in applications that tie applications to a particular data center location. They want to ensure this doesn't happen in the future.

fitness function:

Deterministic Distributability

- Require that developers use configuration files for all database configuration (using tools like ArchUnit)
- During testing, randomly switch database configurations between tests across several working instances

fitness function:

Degrade Gracefully

PenultimateWidgets is proud of how popular their website has become and want to make sure the site performance degrades gracefully as scale increases.

fitness function:

Maintaining Good Throughput

PenultimateWidgets is proud of how popular their website has become and want to make sure they maintain good throughput: a combination of performance and scalability.

fitness function:

Degrade Gracefully

- build a run-time monitor that reports on number of instances, performance per instance and overall
- if performance or scalability violates the established thresholds, raise an alarm

Concrete Architecture Definitions

maintainable?

Concrete Architecture Definitions

maintainable?

Degrade gracefully versus Maintain Throughput ?

fitness function:

RespondResponsibly

PenultimateWidgets business analysts are worried about one of their key competitors, who has a very fast website. PenultimateWidgets website has traditionally not featured speed; they want to ensure that the responsiveness doesn't get worse.

fitness function:

RespondResponsibly

- Spin up a build in an integration environment with representative data
- Use an automated testing tool (such as Selenium or Watir) to automate visiting the pages with an SLA
- Time the rendering of the page (using a tool such as Watchtower)
- Fail the build if the response time exceeds the threshold

fitness function:

Monitor All The Things

PenultimateWidgets wants to ensure that Java and .NET applications support monitoring.

fitness function:

Monitor All The Things

- create a test during the pre-flight building of the machine image to ensure monitoring ports are listening properly
- fail the build if any development team skipped or mis-configured monitoring

Anemic BA Anti-pattern

Product Owner

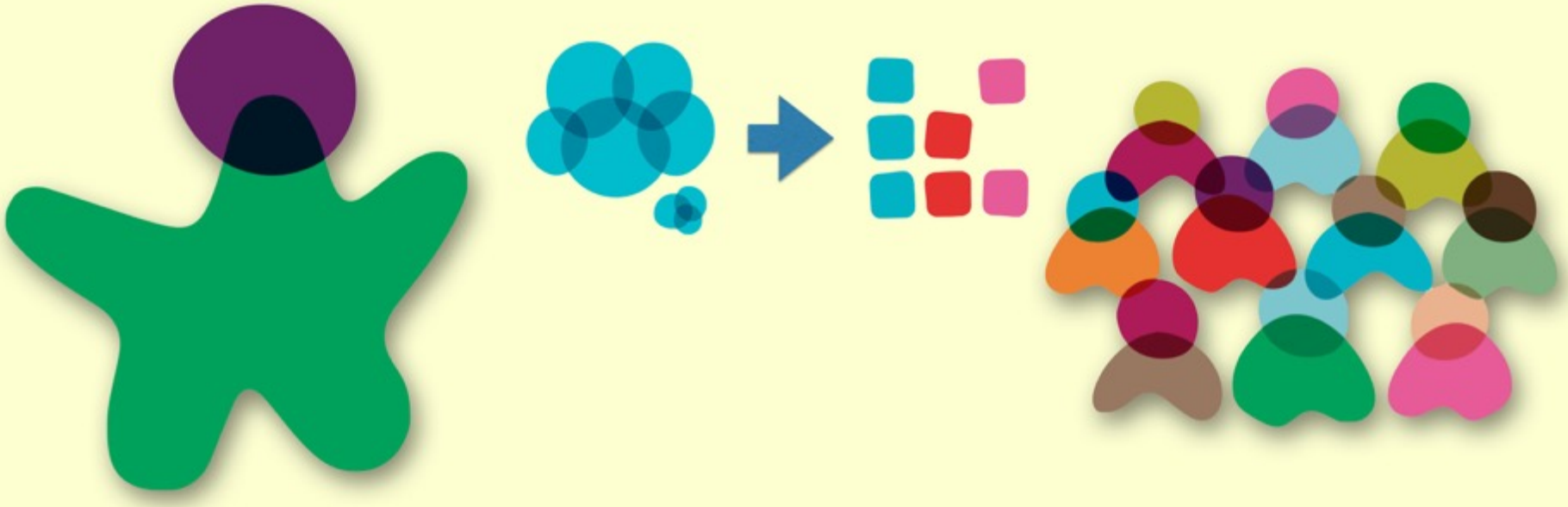
Developers

Anemic BA Anti-pattern

Product Owner

Developers

Anemic BA Anti-pattern



Product Owner

Developers

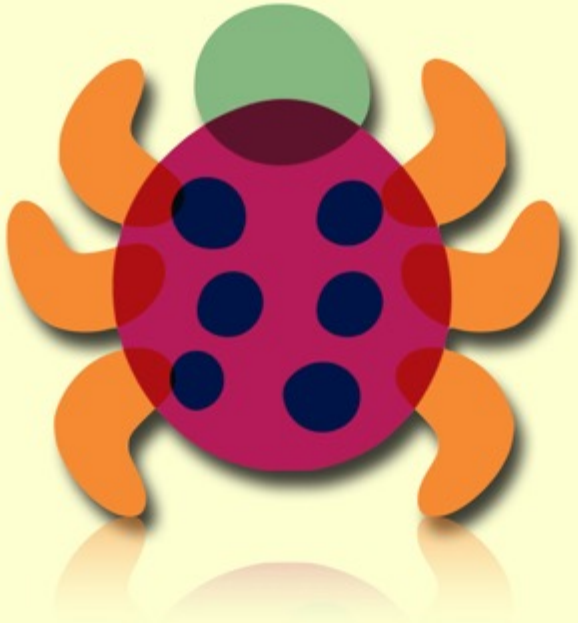
Anemic BA Anti-pattern

Product Owner

Business Analysts

Developers

Breaking Bad Habits



fitness function:

DireDeltas

PenultimateWidgets developers have developed a bad habit of keeping files checked out for days or weeks at a time, preventing continuous integration. As part of their attempt to improve engineering practices overall, the enterprise architects at PenultimateWidgets encourage frequent check-ins with corresponding good test coverage.

fitness function:

DireDeltas

- Create a fitness function that counts the number of lines in the check-in delta and fails if it exceeds the threshold
- Create a fitness function that counts the number of lines in delta from the test packages and the source packages and compute the ratio; fail if developers exceed the defined ratio

fitness function:

DireDeltas

good fitness function?

fitness function:

DireDeltas

- Create a fitness function that counts the number of lines in the check-in delta and fails if it exceeds the threshold
- Create a fitness function that counts the number of lines in delta from the test packages and the source packages and compute the ratio; fail if developers exceed the defined ratio

How would you make it better?

fitness function:

No More View Models

PenultimateWidgets has a group of developers who cut their teeth on mainframe programming, where all work is done via a terminal. They have (mostly) switched to modern web development, but architects find that they often create models and workflows directly in web pages rather than support a cleaner MVC separation, and they would like to discourage this behavior.

fitness function:

No More View Models

- use PMD (https://pmd.github.io/) or Language server protocol (https://langserver.org/) to build a tool that scans which classes developers instantiate in view code
- build a list of restricted model and controller packages
- if any view code instantiates a class from the restricted list, fail the build

fitness function:

Deploy All The Things

PenultimateWidgets has gotten tired of each application requiring an elaborate environment setup to deploy and run, and has decided that all new applications must be deployed within Docker containers. The PenultimateWidgets architects want to ensure that all applications follow the same standard deployment guidelines.

fitness function:

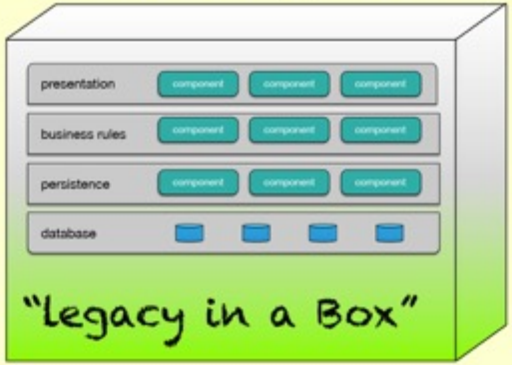
Deploy All The Things

Build a monitor that runs within the developers' staging area that inspects each running service to ensure that it is running within a container and supports common operational concerns such as logging, monitoring, and service discovery.

Penultimate fitness function:

Deploy All The Things

Build a monitor that runs within the developers' staging area that inspects each running service to ensure that it is running within a container and supports common operational concerns such as logging, monitoring, and service discovery.



Make New Habits

fitness function:

Discover All The Things

PenultimateWidgets publishes an API for third-party, business-to-business integrations. To make it easier for clients to consume their services, they have instituted a convention that every endpoint delivers a documentation string when called with the additional parameter '&docs=true'. PenultimateWidgets architects want to ensure that all published endpoints include this documentation string.

Penultimate fitness function:

Discover All The Things

Developers build a continuous monitor that calls each method in the published API at random from a dictionary, checking that each endpoint documentation invocation delivers an HTML page and not an HTML 404.

fitness function:

Install All The Things

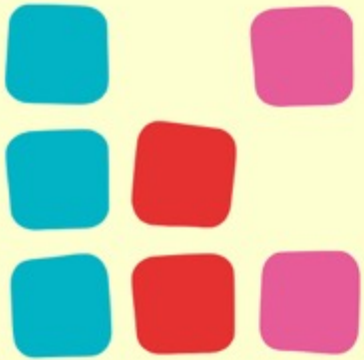
PenultimateWidgets distributes a desktop application to select clients that facilitates coordination between business partners. The application is written in C#, using .NET desktop libraries. In the past, PenultimateWidgets has exhibited problems with the installer not working correctly for some edge cases, and they want to ensure that their clients have a good installation experience.

fitness function:

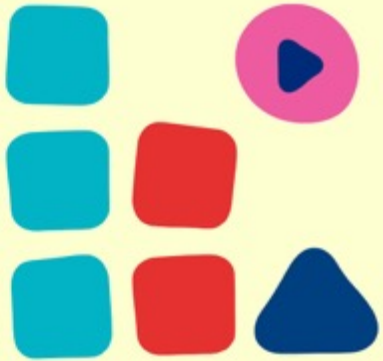
Install All The Things

- As part of the deployment pipeline, provision a pristine Windows environment for each supported version
- Install the new build and run the installation routine
- Smoke test select parts of the application to ensure that all components installed correctly.

Aggressive Back Porting



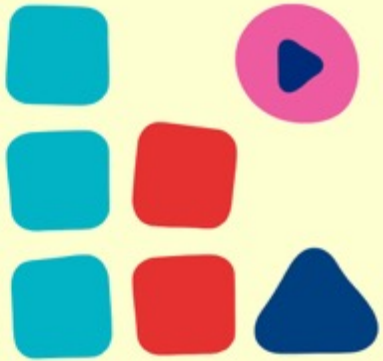
Version X



Version X+1

Aggressive Back Porting

Version X



Version X+1

fitness function:

Break on Upgrade

PenultimateWidgets builds a number of applications in Ruby on Rails, which has an active release schedule. Occasionally, developers really want a feature that appears in version X+1, where PenultimateWidgets is currently on version X. To acquire the desired functionality, developers back-port the new features from X + 1 into X. However, when PenultimateWidgets finally does upgrade to X + 1, the back ported features are almost certain to break because of incompatibilities.

fitness function:

Break on Upgrade

- Add a unit test to the code base that checks the version number of the framework. If the framework version it higher than the expected one, fail the build with a warning that back-ported features will likely not work.
- Catalog all the back-ported features in the unit test to make it easy to identify them when it becomes time to replace them.

fitness function:

Upgrade All The Things

PenultimateWidgets has a bad habit of waiting too long to update core libraries and frameworks they depend upon for development. Waiting past several versions makes the eventual upgrade quite painful, and they have resolved to perform upgrades in a more timely manner.

fitness function:

Upgrade All The Things

- Use Dependabot to flag new major versions of frameworks and libraries.
- When a library updates, make an entry in a build database and start a 3-month clock.
- If the library or framework in question hasn't been updated in 3 months, fail the build.

fitness function:

Depend on Dependencies

PenultimateWidgets has recently been burned by a Maven update that broke several applications yet wasn't detected until production. Architects and business people want to make sure this doesn't happen again.

fitness function:

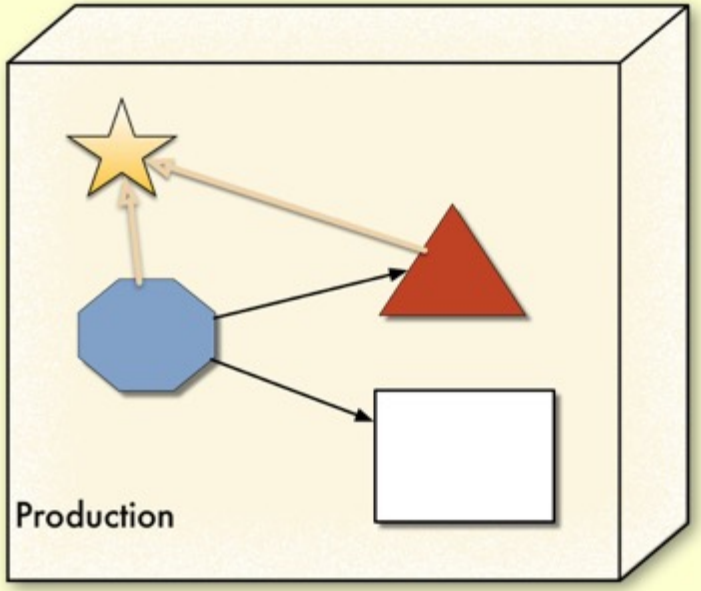
Depend on Dependencies

Use a deployment pipeline to trigger new builds and test suite execution any time any artifact of the project changes, including external dependencies. If the automatic build breaks, a developer must locate the problem and restore automatically.

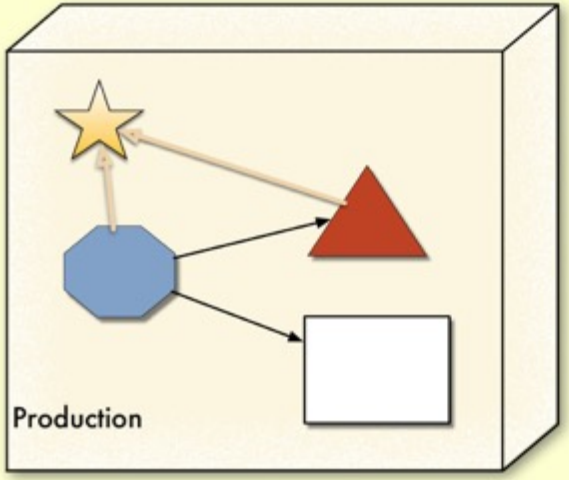
OR

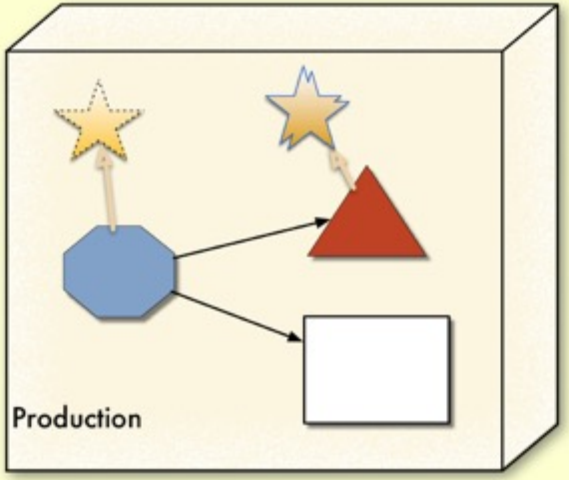
Use Dependabot (https://dependabot.com/)

Penultima⬆e



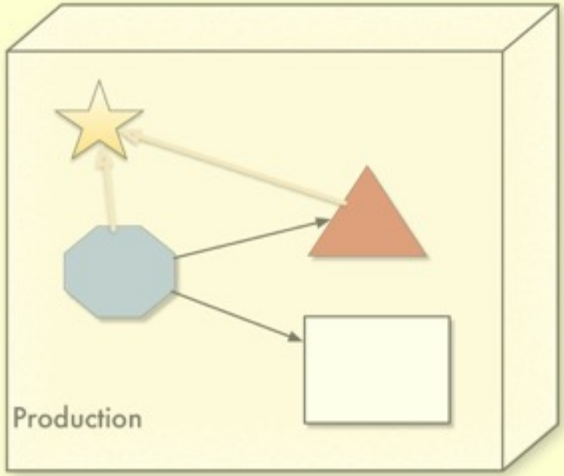
Penultima⬆e

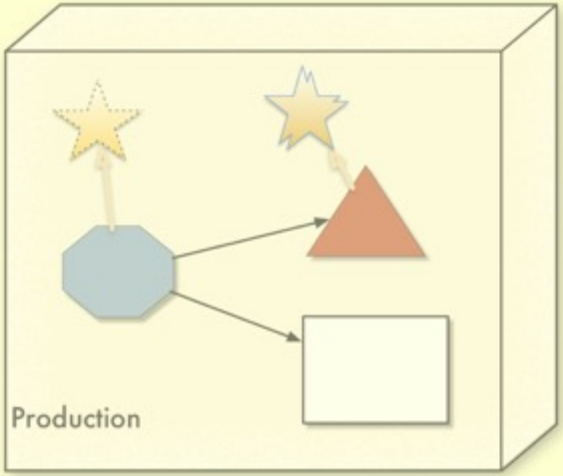


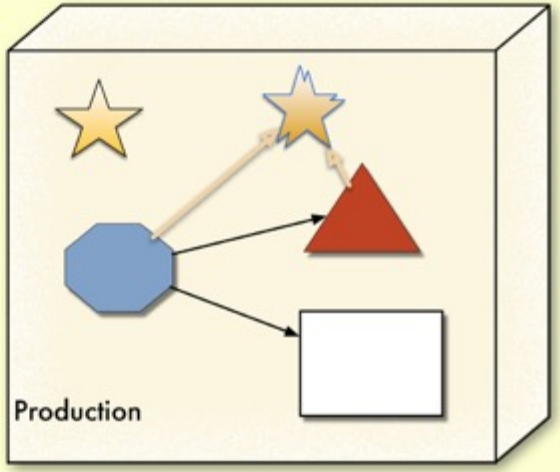


Penultimate

incremental







Penultimate

Evolving Routing

< >

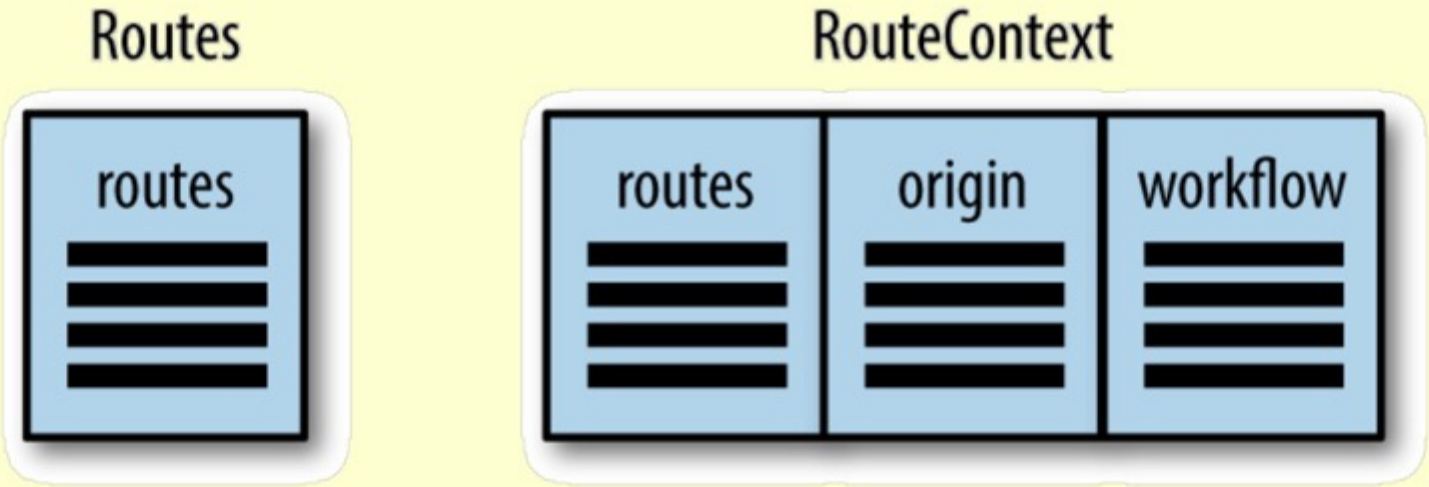
<home> <catalog><item>

Penultimate

Evolving Routing

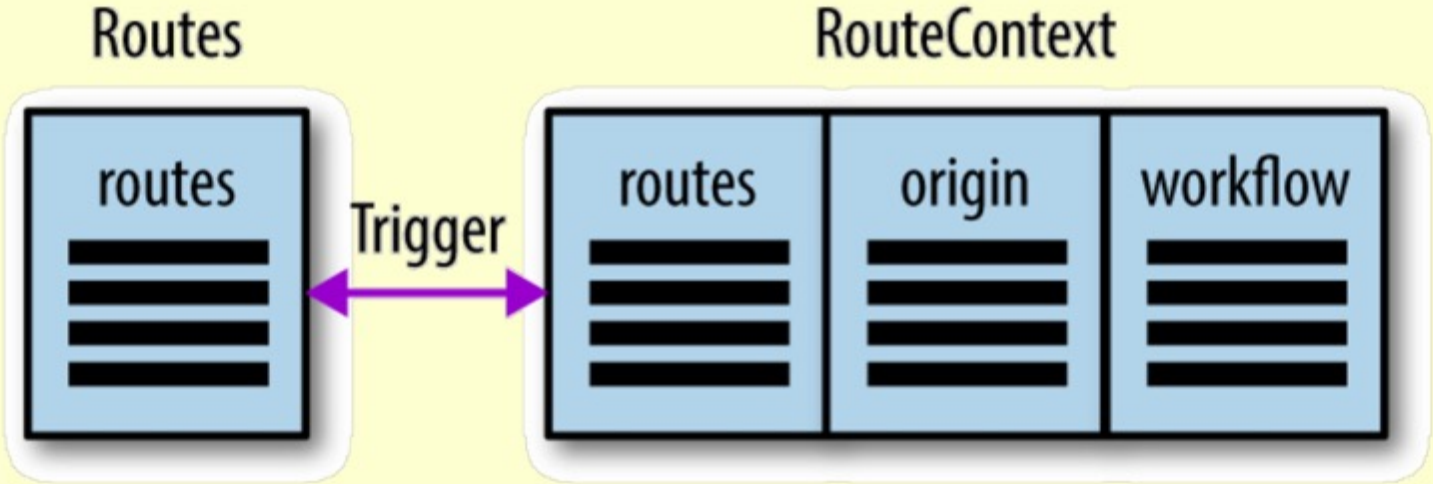
Penultimate

Evolving Routing



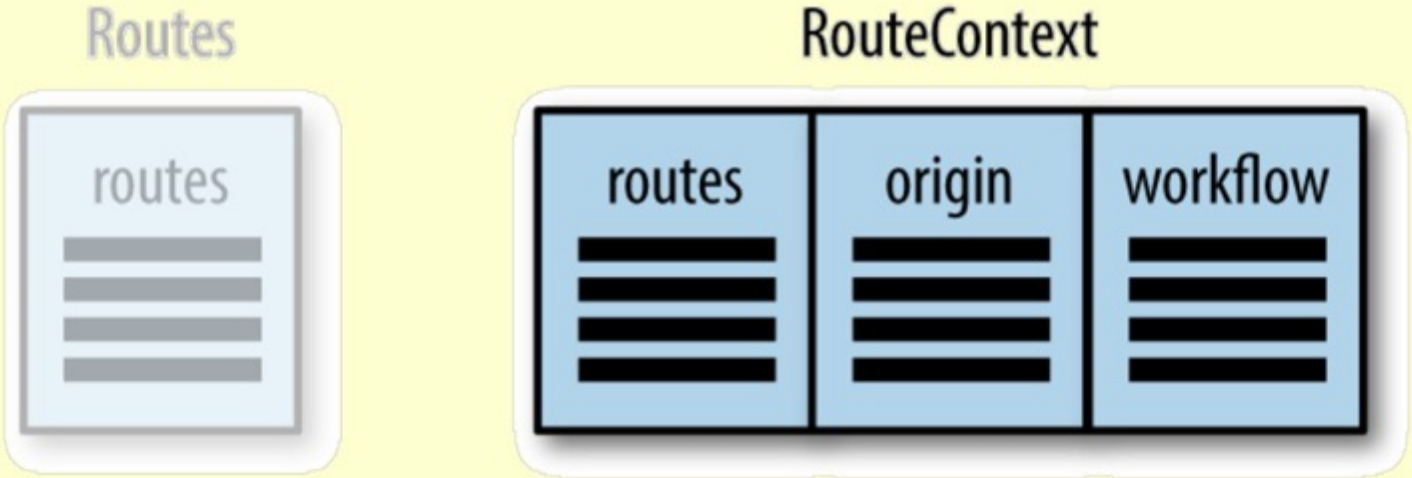
Penultimate

Evolving Routing



Penultimate

Evolving Routing



Penultimate

Experimenting with New Markets

fitness function:

MaintainTheMock

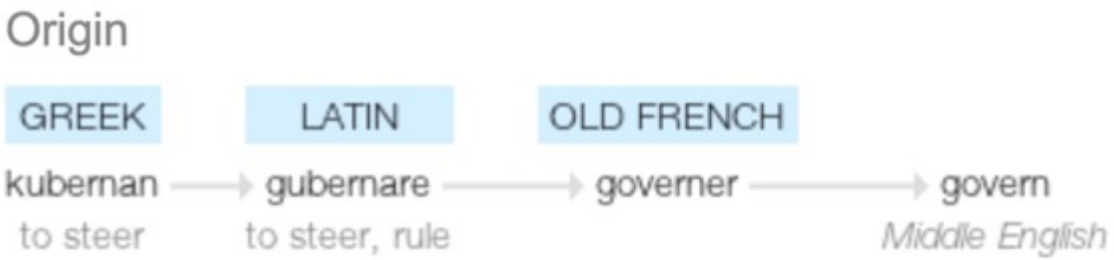
PenultimateWidgets has started experimenting with flashing the chips on their 'smart' widgets during the manufacturing process, allowing them to make changes in hardware-based code more easily and quickly. However, they must make sure that the mock version of the hardware API the developers use is always up to date with the real API.

fitness function:

MaintainTheMock

Build a fitness function that matches the mock API method signatures, parameters, and types to the real API on the hardware; fail the build if the team has allowed it to get out of sync.

govern

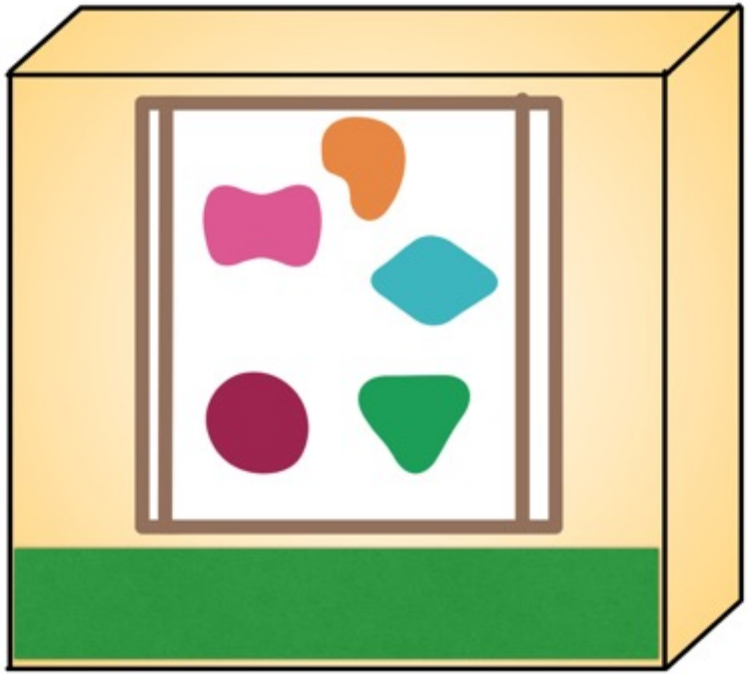


— to control, direct, or strongly influence the actions and conduct of

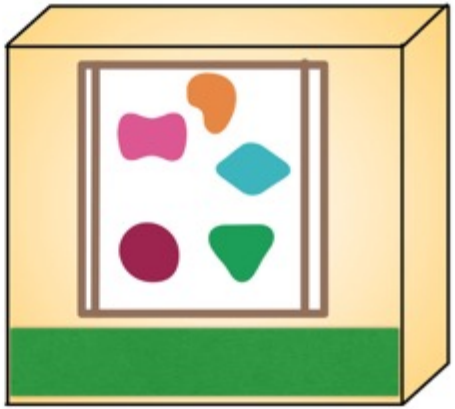
— to exert a determining or guiding influence in or over

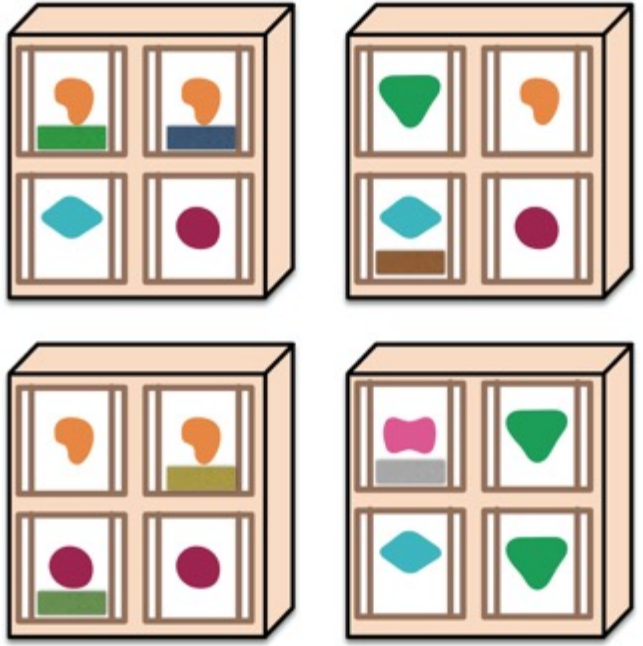
— to hold in check

monolithic governance

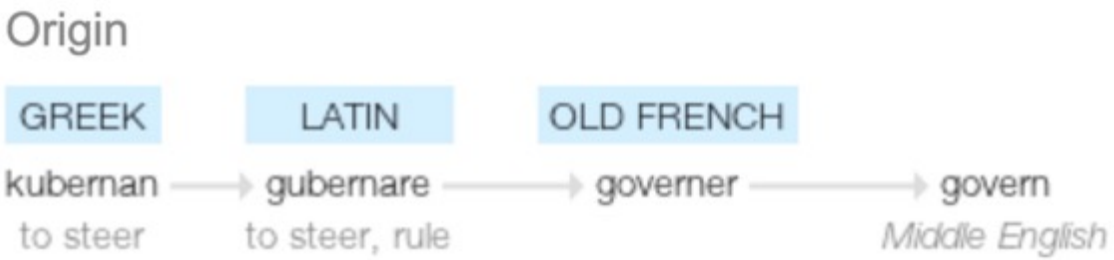


decentralized governance





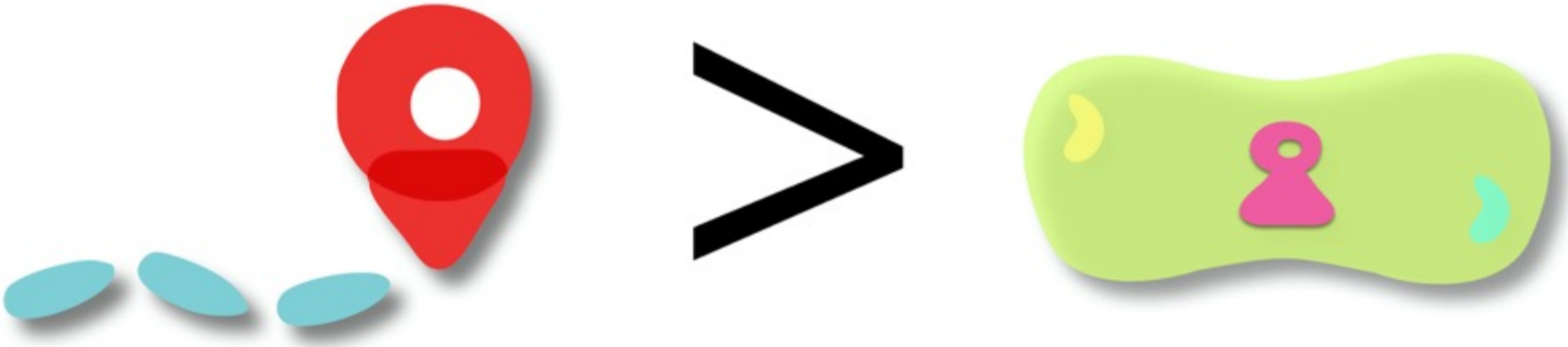
govern



— to serve as a precedent or deciding principle for

20th Century Governance

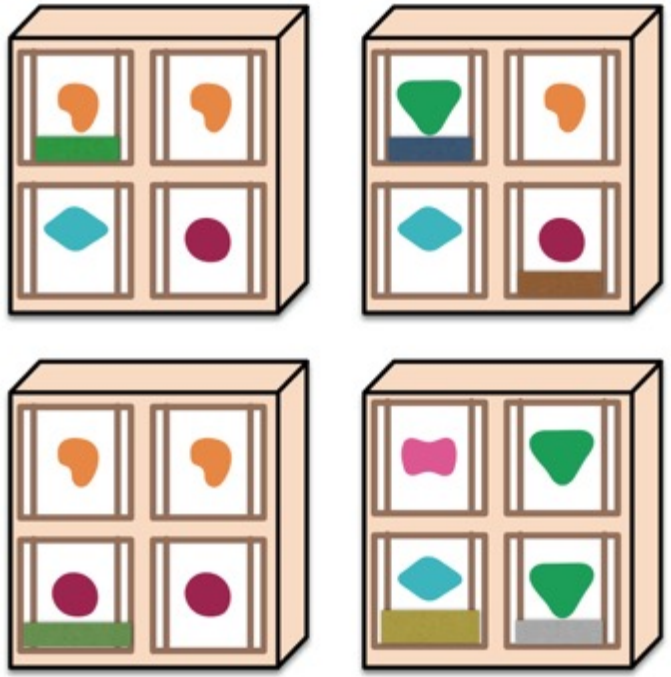
21st Century Governance



guidance over cost savings

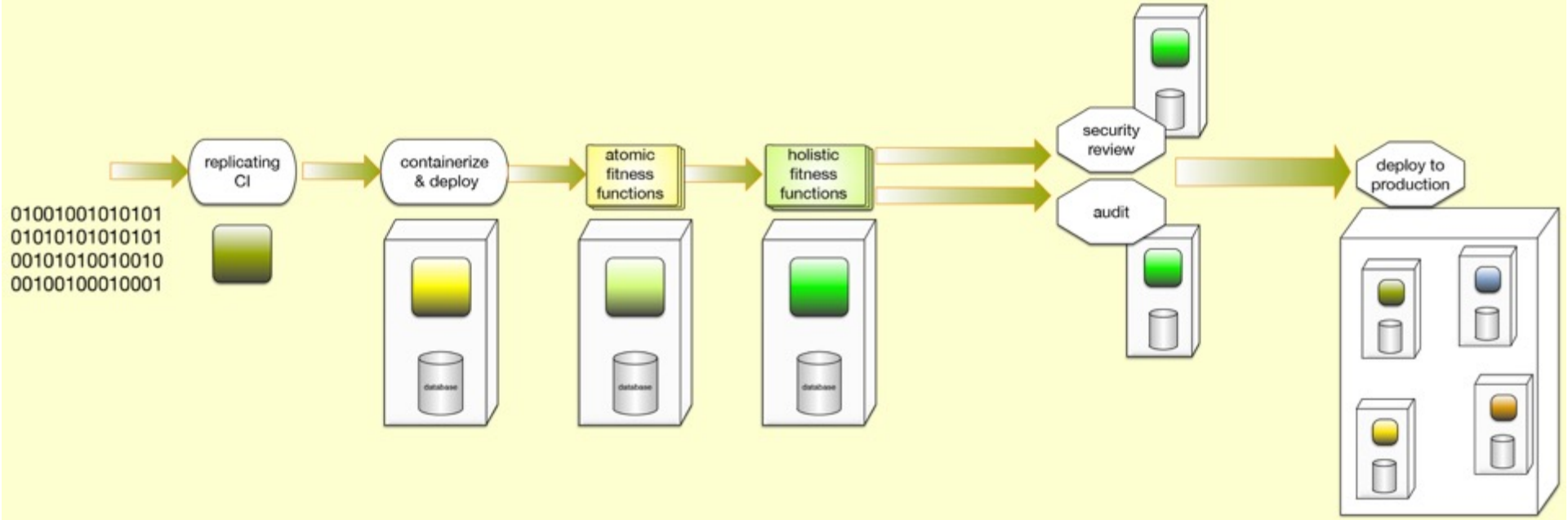
"Goldilocks" Governance

Choose technology stacks appropriate to problem scale.



Penultimate

Deployment Pipeline



fitness function:

Zero Day Security Check

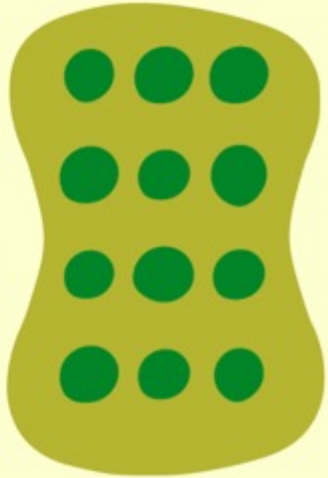
PenultimateWidgets management has become concerned about zero-day exploits in open source libraries and has tasked the security team with coming up with a way of ensuring that projects do not use known compromised versions.

fitness function:

Zero Day Security Check

- build a common fitness function into every deployment pipeline for the security team
- have the security fitness function check version numbers of deployed libraries
- if a tainted version is in use, fail the build and notify the team

Penultima e



fitness function:

Doc Sync With API

PenultimateWidgets publishes an API for external vendors to check on order status, availability, and other B2B perks. However, the external partners rely on the documentation of the API, which sometimes becomes outdated when developers add new functionality. The PenultimateWidgets architects want to ensure that documentation is always up to date, especially method signatures and data formats.

fitness function:

Doc Sync With API

- Parse the code to look for public method signatures
- Parse the documentation looking for <code> tags, which surround method signatures and parameters
- Ensure that each tagged element includes a description
- Add a Publication Date value to the public website; after new deployments, check to make sure the deployment time matches the Publication date

fitness function:

Sell The Platform

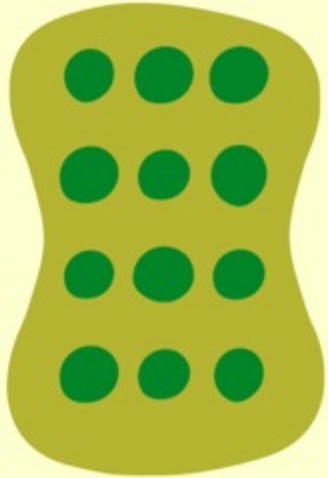
PenultimateWidgets is in the process of rebuilding their core platform, incorporating some key features that will appeal to others in a similar problem space. PenultimateWidgets plans to sell their platform to other companies, but worry that the differentiating factors (such as performance, scale, and elasticity) will suffer as clients start making changes to the platform, ultimately harming PenultimateWidgets platform's reputation. How can they sell their platform and still ensure that the critically important characteristics remain unchanged?

fitness function:

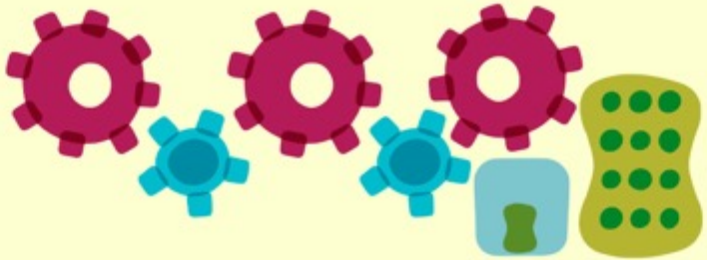
Sell The Platform

- PenultimateWidgets sells not only the platform but the running deployment pipeline for the platform as well. PenultimateWidgets engineers make sure that all the fitness functions run in the new client's environment.
- PenultimateWidgets will continue to certify the platform as long as all their fitness functions continue to execute successfully.

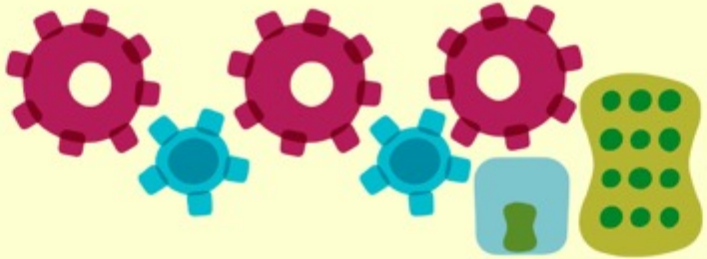
Penultima e



Penultima e

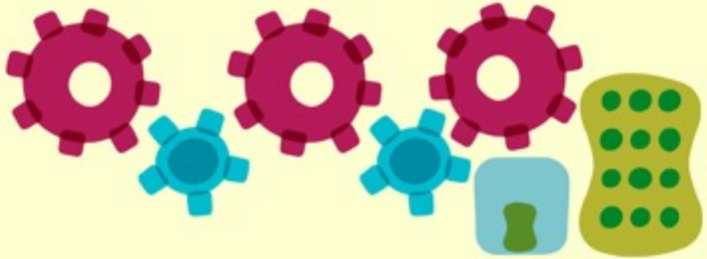


Penultimate



alien artifacts!

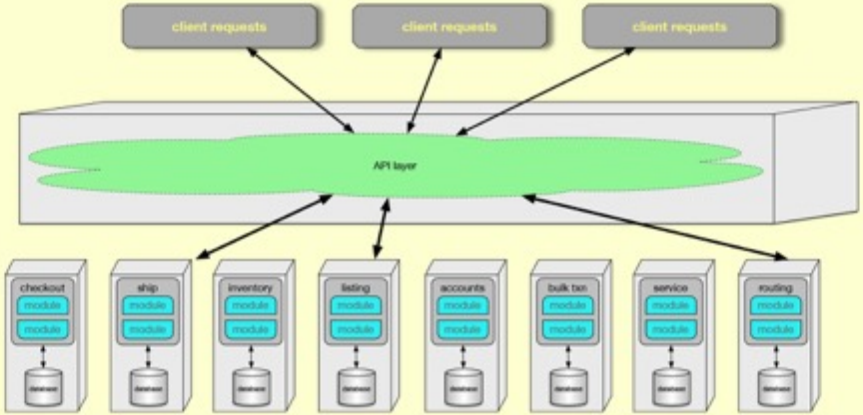
Penultimate



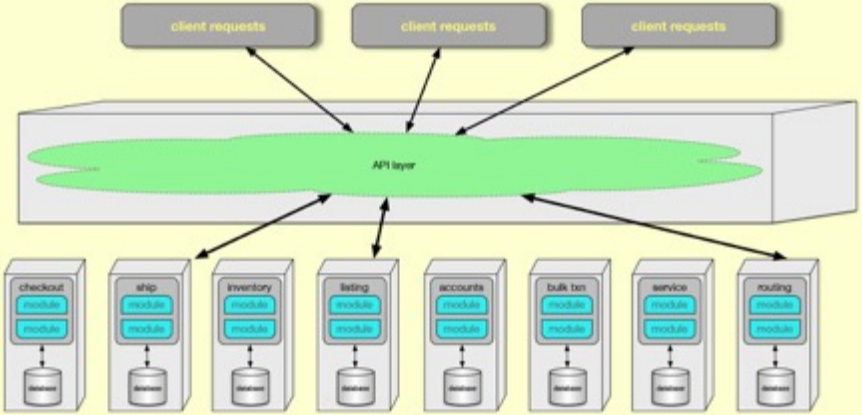
alien artifacts!

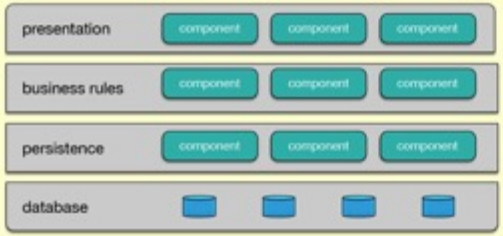
integration architecture

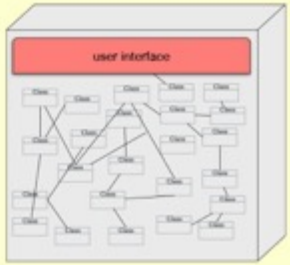
(Forced) Integration

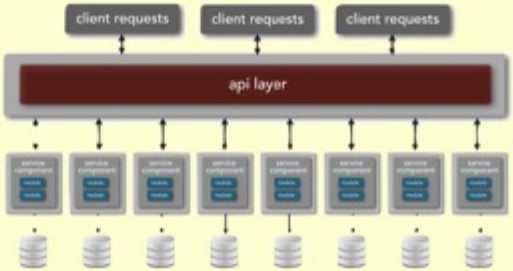


(Forced) Integration

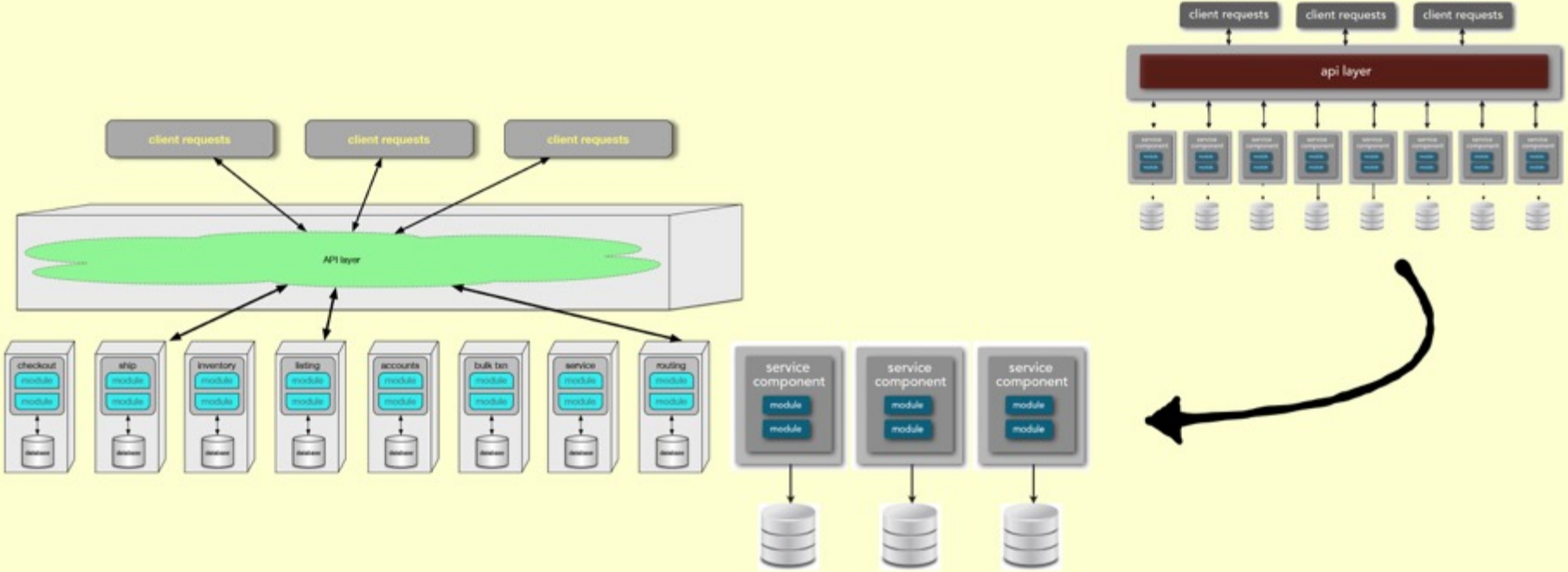




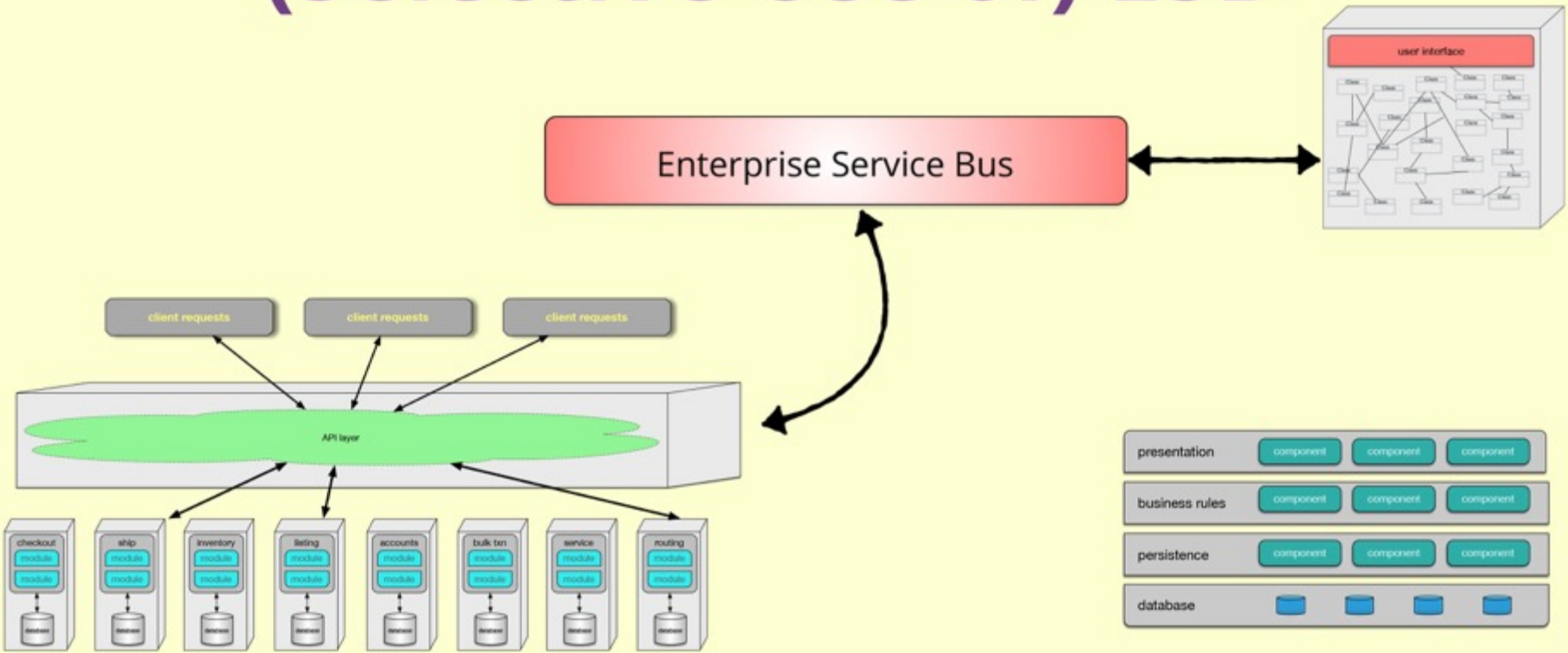




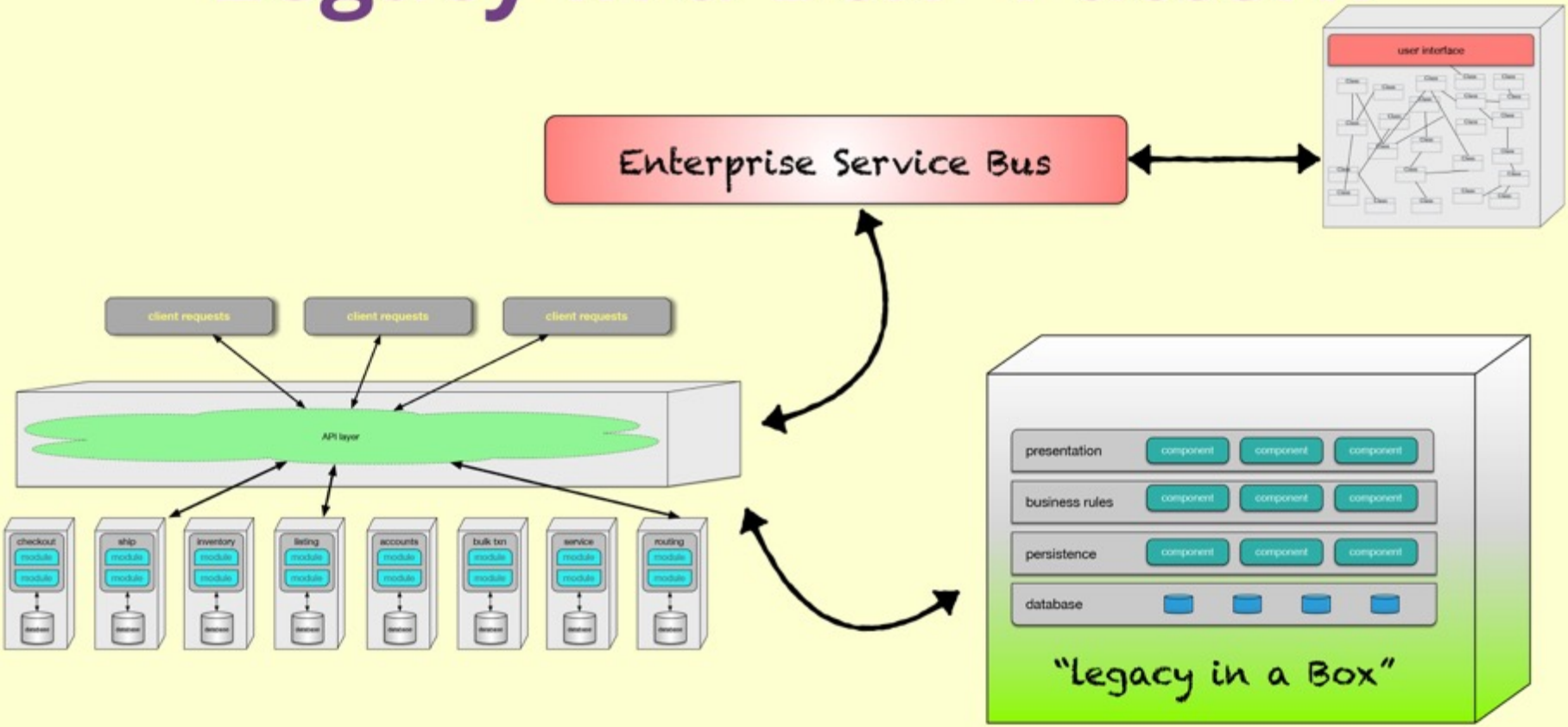
(Forced) Integration



(Selective Use of) ESB

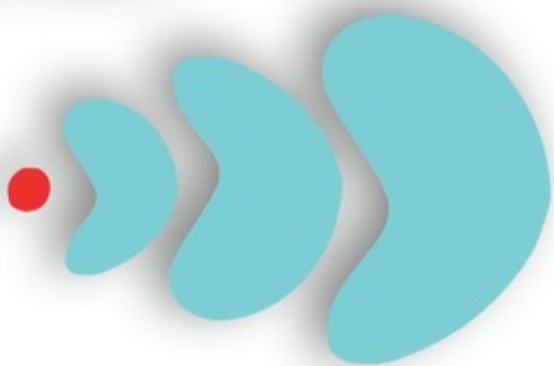


“Legacy in a Box” Pattern



puBLisHinG goVeRnANce

Build Your Own Technology Radar



ThoughtWorks®

Technology Radar

OUR THOUGHTS ON JAVASCRIPT, APIS, CONWAY'S LAW, RE-DECENTRALIZATION AND MUCH MORE

JULY 2014

thoughtworks.com/radar

THE RADAR

TECHNIQUES

ADOPT
1 Capturing client-side JavaScript errors
2 Continuous delivery for mobile devices
3 Mobile testing on mobile networks
4 Segregated OOM plus node for JS Testing
5 Windows Infrastructure automation

TRIAL
6 Capture domain events explicitly
7 Client and server rendering with same code
8 HTML5 storage instead of cookies
9 Instrument all the things
10 Masterless Chef/Puppet
11 Micro-services
12 Perimeterless enterprise
13 Provisioning testing
14 Structured Logging

ASSESS
15 Bridging physical and digital worlds with simple hardware
16 Collaborative analytics and data science
17 Datensparsamkeit
18 Development environments in the cloud
19 Focus on mean time to recovery
20 Machine image as a build artifact
21 Tangible interaction

HOLD
22 Cloud lift and shift
23 Ignoring OWASP Top 10
24 Siloed metrics
25 Velocity as productivity

PLATFORMS

ADOPT
26 Elastic Search
27 MongoDB
28 Neo4j
29 Node.js
30 Redis
31 SMS and USSD as a UI

TRIAL
32 Hadoop 2.0
33 Hadoop as a service
34 OpenStack
35 PostgreSQL for NoSQL
36 Vumi

ASSESS
37 Akka
38 Backend as a service
39 Low-cost robotics
40 PhoneGap/Apache Cordova
41 Private Clouds
42 SPDY
43 Storm
44 Web Components standard

HOLD
45 Big enterprise solutions
46 CMS as a platform
47 Enterprise Data Warehouse

THE RADAR

TOOLS

ADOPT
48 D3
49 Dependency management for JavaScript

TRIAL
50 Ansible
51 Calabash
52 Chaos Monkey
53 Gatling
54 Grunt.js
55 Hystrix
56 Icon fonts
57 Librarian-puppet and Librarian-Chef
58 Logstash & Graylog2
59 Moco
60 PhantomJS
61 Prototype On Paper
62 SnapCI
63 Snowplow Analytics & Piwik

ASSESS
64 Cloud-init
65 Docker
66 Octopus
67 Sensu
68 Travis for OSX/iOS
69 Visual regression testing tools
70 Xamarin

HOLD
71 Ant
72 Heavyweight test tools
73 TFS

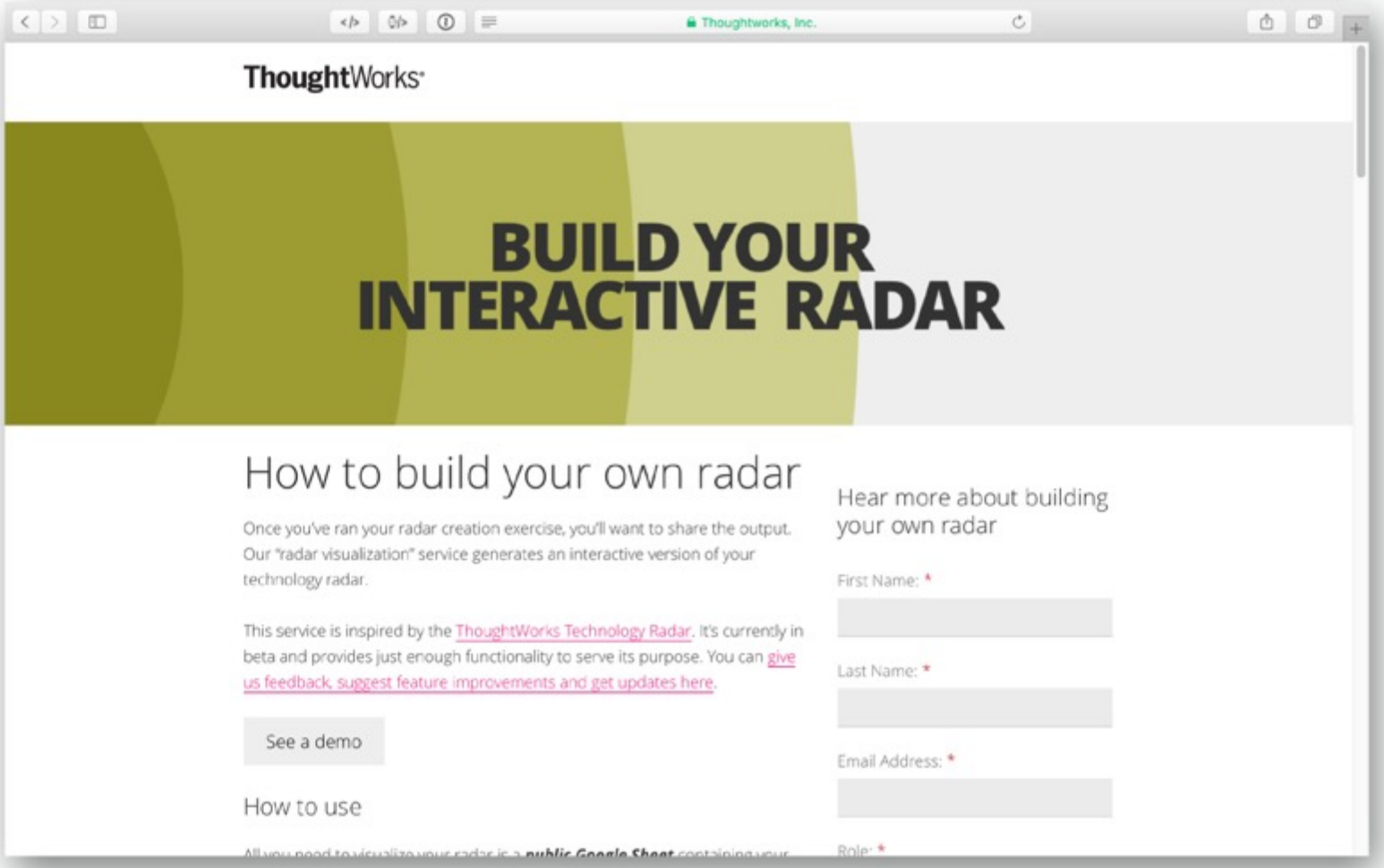
LANGUAGES & FRAMEWORKS

ADOPT
74 Clojure
75 Dropwizard
76 Scala, the good parts
77 Sinatra

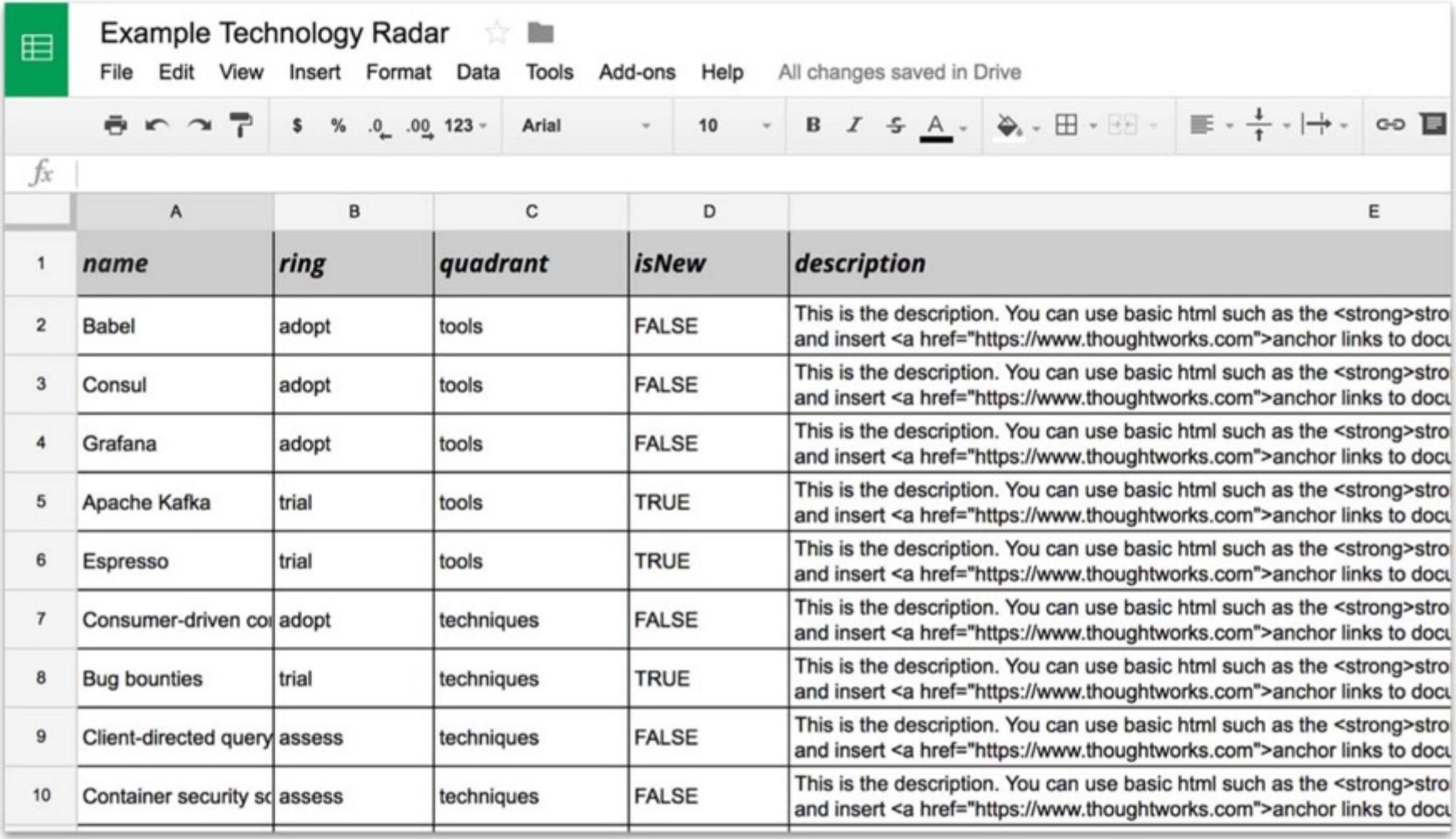
TRIAL
78 CoffeeScript
79 Go language
80 Hive
81 Play Framework 2
82 Reactive Extensions across languages
83 Web API

ASSESS
84 Elixir
85 Julia
86 Nancy
87 OWIN
88 Pester
89 Pointer Events
90 Python 3
91 TypeScript
92 Yesman

HOLD
93 Handwritten CSS
94 JSF

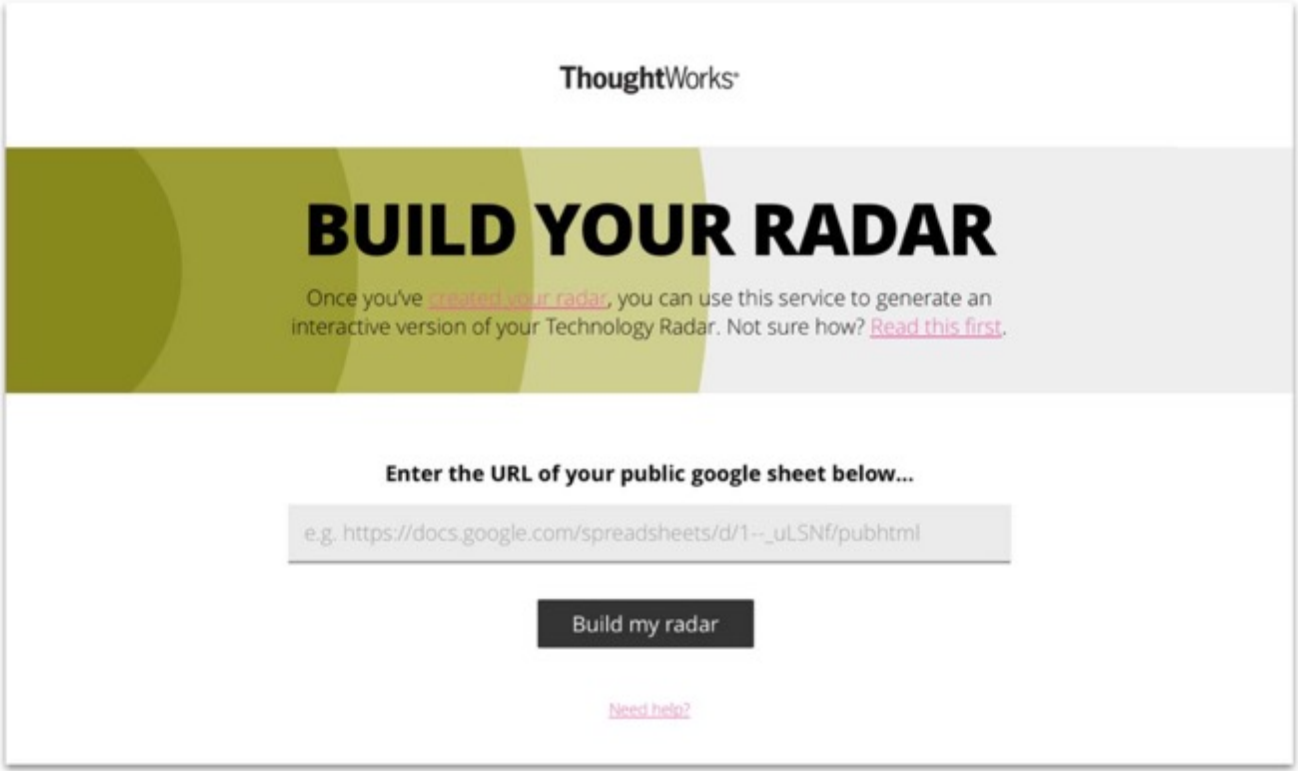


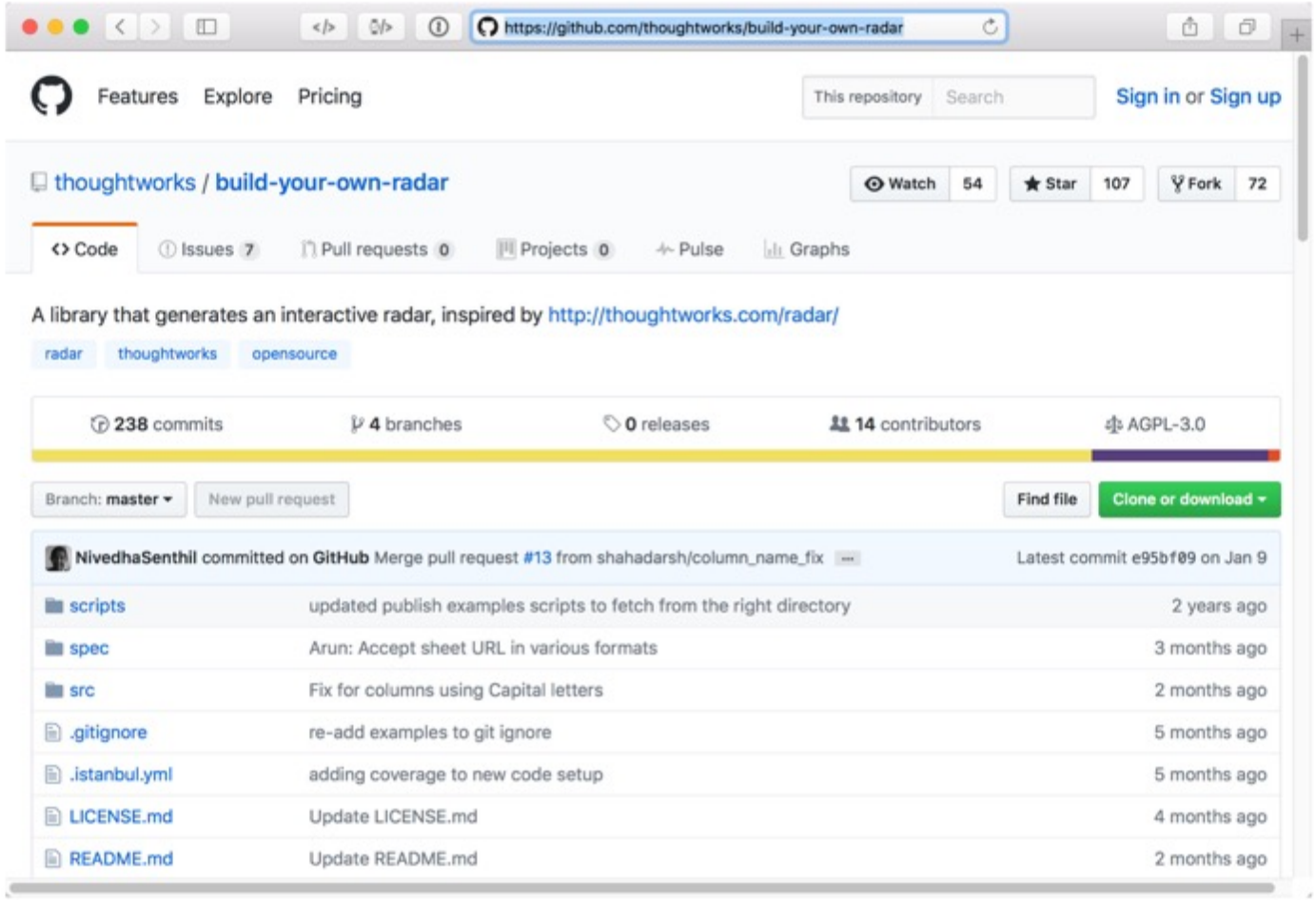
https://info.thoughtworks.com/visualize-your-tech-strategy-guide.html


Example Technology Radar

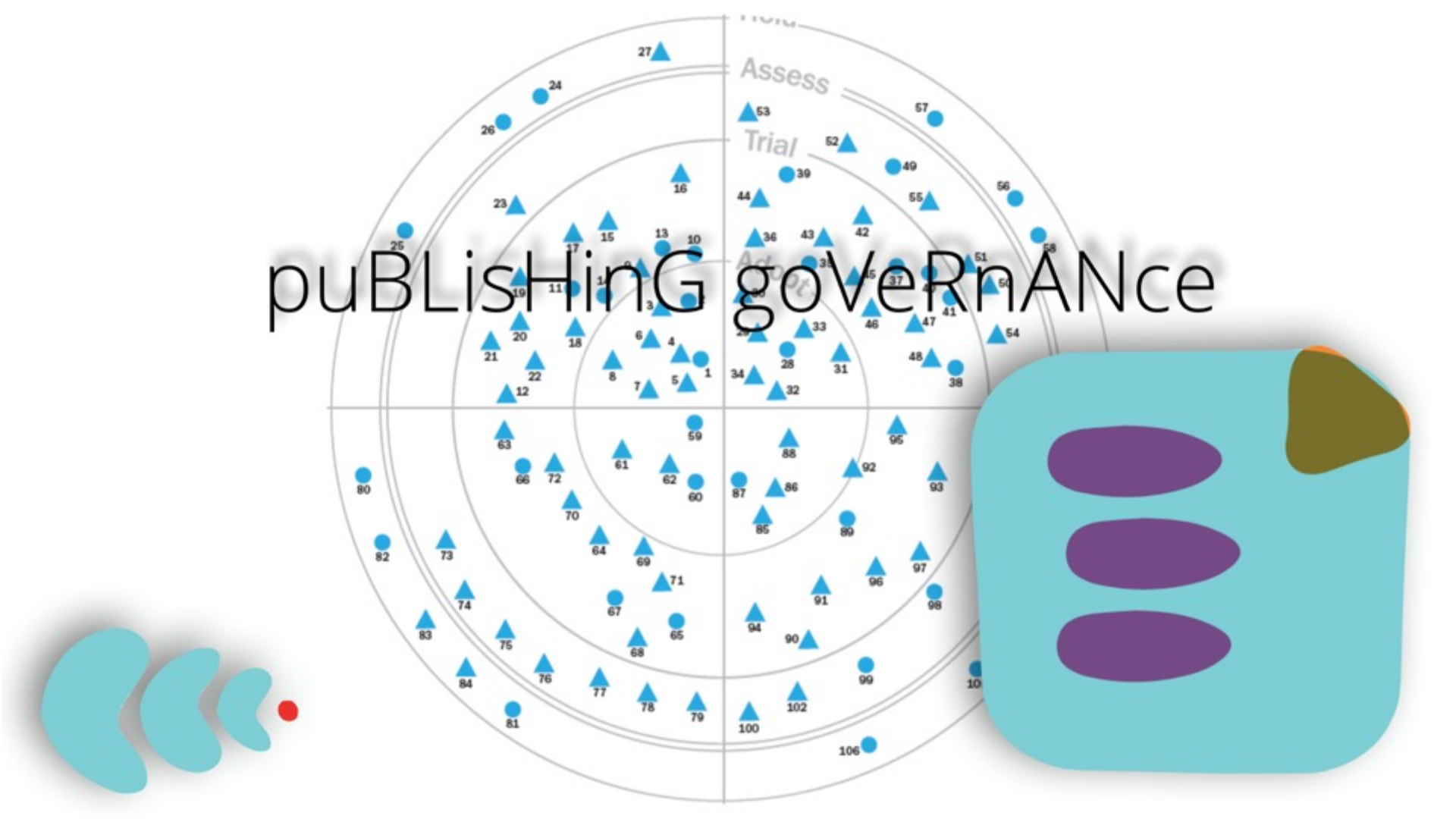
File Edit View Insert Format Data Tools Add-ons Help All changes saved in Drive

	A	B	C	D	E
1	***name***	***ring***	***quadrant***	***isNew***	***description***
2	Babel	adopt	tools	FALSE	This is the description. You can use basic html such as the stro and insert <a href="https://www.thoughtworks.com">anchor links to docu
3	Consul	adopt	tools	FALSE	This is the description. You can use basic html such as the stro and insert <a href="https://www.thoughtworks.com">anchor links to docu
4	Grafana	adopt	tools	FALSE	This is the description. You can use basic html such as the stro and insert <a href="https://www.thoughtworks.com">anchor links to docu
5	Apache Kafka	trial	tools	TRUE	This is the description. You can use basic html such as the stro and insert <a href="https://www.thoughtworks.com">anchor links to docu
6	Espresso	trial	tools	TRUE	This is the description. You can use basic html such as the stro and insert <a href="https://www.thoughtworks.com">anchor links to docu
7	Consumer-driven co	adopt	techniques	FALSE	This is the description. You can use basic html such as the stro and insert <a href="https://www.thoughtworks.com">anchor links to docu
8	Bug bounties	trial	techniques	TRUE	This is the description. You can use basic html such as the stro and insert <a href="https://www.thoughtworks.com">anchor links to docu
9	Client-directed query	assess	techniques	FALSE	This is the description. You can use basic html such as the stro and insert <a href="https://www.thoughtworks.com">anchor links to docu
10	Container security sc	assess	techniques	FALSE	This is the description. You can use basic html such as the stro and insert <a href="https://www.thoughtworks.com">anchor links to docu


ThoughtWorks
BUILD YOUR RADAR
Once you've created your radar, you can use this service to generate an interactive version of your Technology Radar. Not sure how? Read this first.
Enter the URL of your public google sheet below...
e.g. https://docs.google.com/spreadsheets/d/1--_uLSNf/pubhtml
Build my radar
Need help?

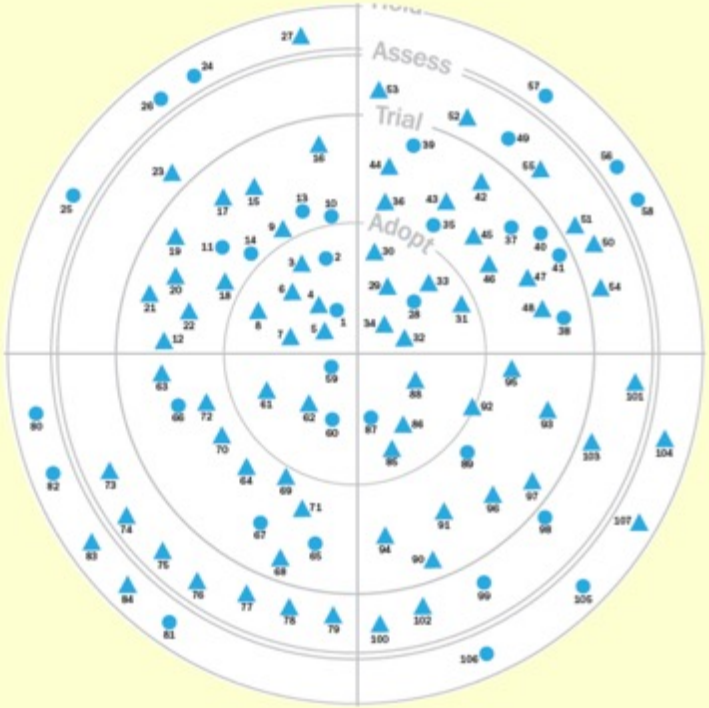


https://github.com/thoughtworks/build-your-own-radar


puBLisHinG goVeRnANce
Assess
Trial

Penultimate

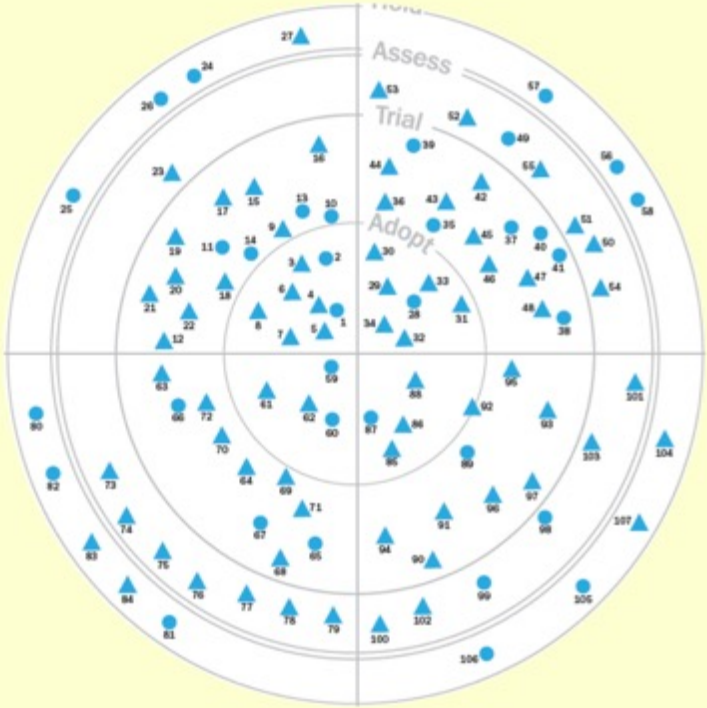
enterprise architecture transparency



Penultimate

Setting WIP limits for experiments

enterprise architecture transparency

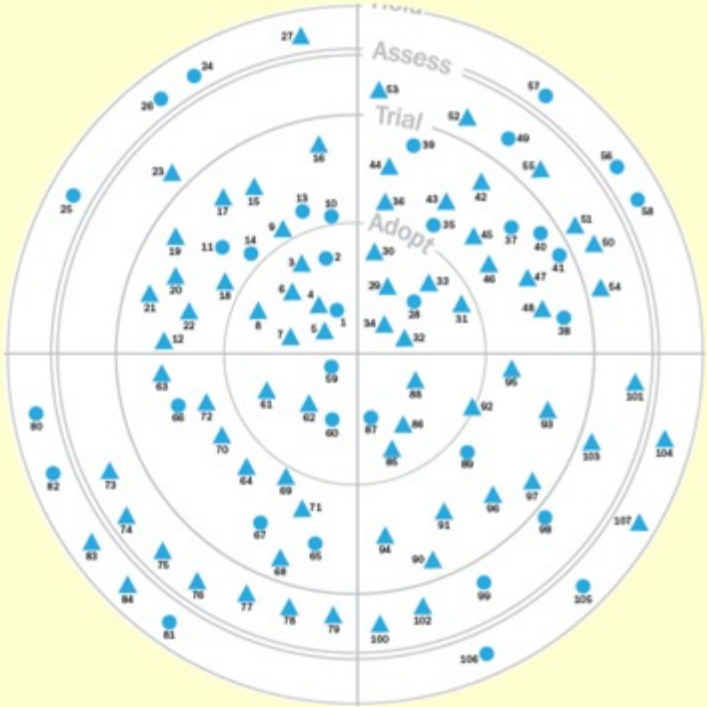


Penultimate

Setting WIP limits for experiments

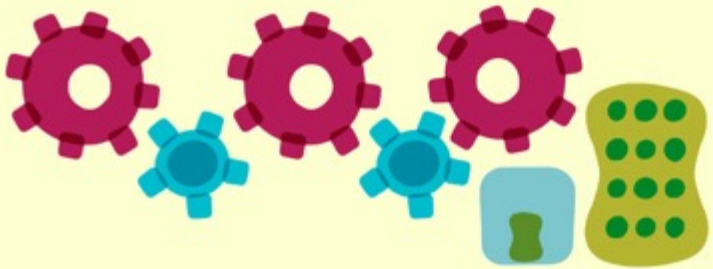
enterprise architecture transparency

Broadcasting R&D

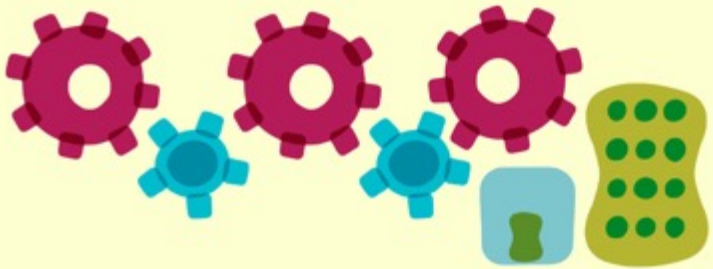


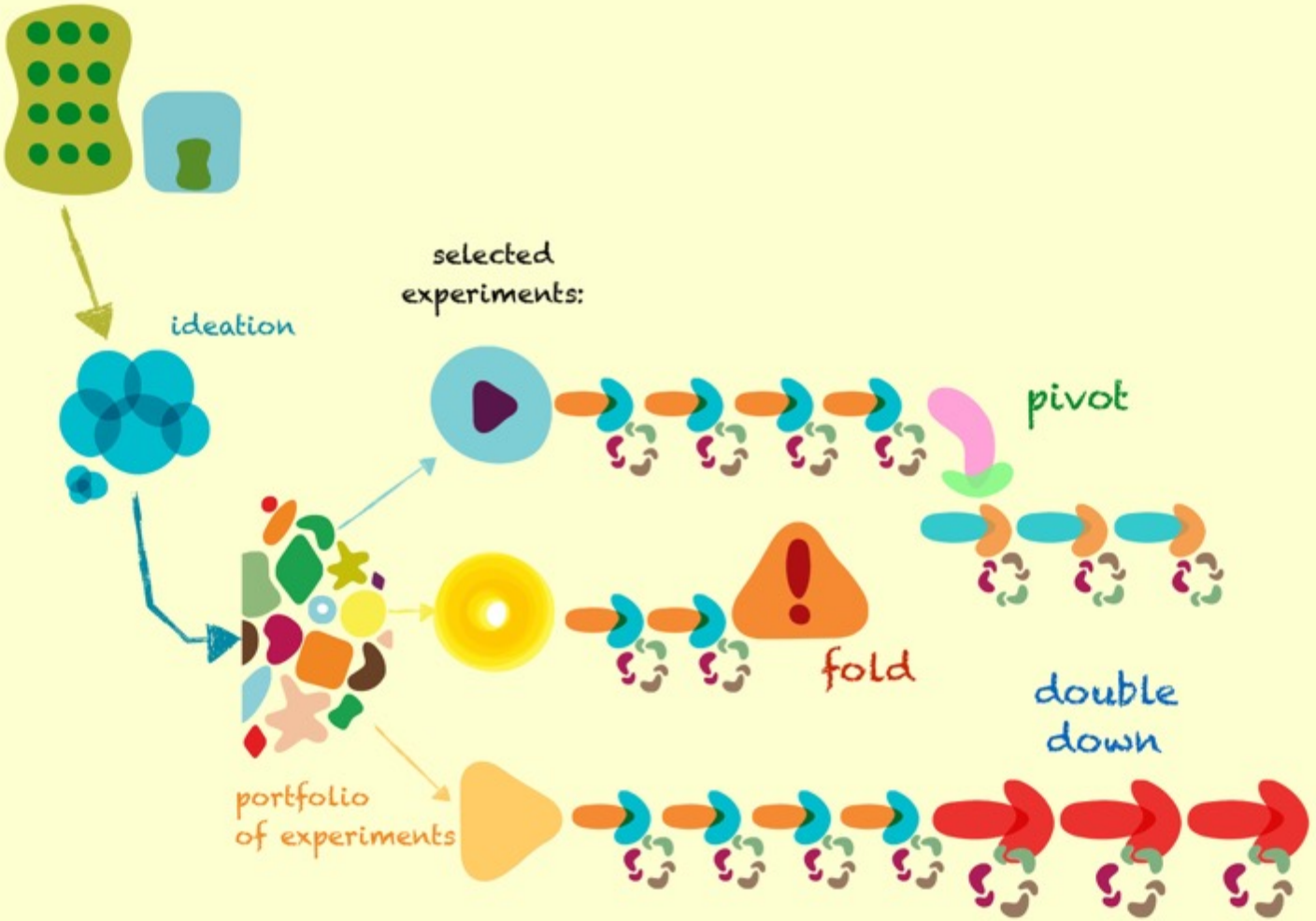
Penultima e

Penultima e

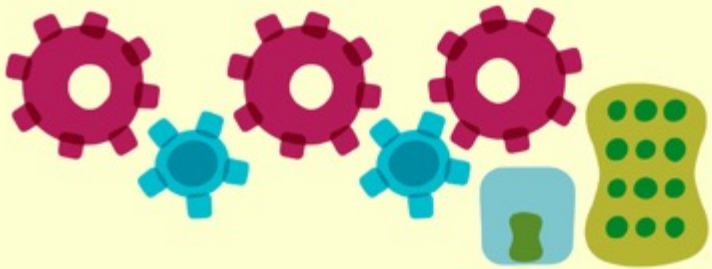


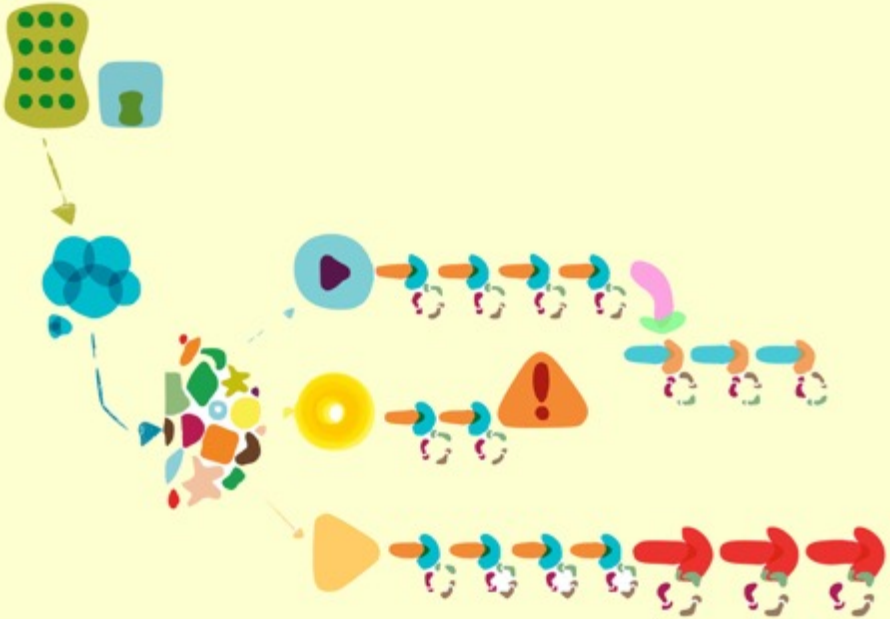
Penultima e




selected
experiments:
ideation
pivot
fold
double
down
portfolio
of experiments

Penultima⬆e






Penultimate
OpenUltimate

Penultima⬆e

OpenUltima⬆e

Evolutionary architecture provides the substrate for 21st century growth.

Fitness Function Guidelines

most fitness function are local

Fitness Function Guidelines

most fitness function are local

global only when universally applied.

Fitness Function Guidelines

most fitness function are local

global only when universally applied.

No EA  states!

Fitness Function Guidelines

developers must collaborate

Fitness Function Guidelines

developers must collaborate

No EA  states!

Fitness Function Guidelines

prefer automation but rely on manual when needed

Fitness Function Guidelines

prefer automation but rely on manual when needed

automation =  when needed

Architecture Defined

architecture =  + 

Architecture Defined

architecture =  + 

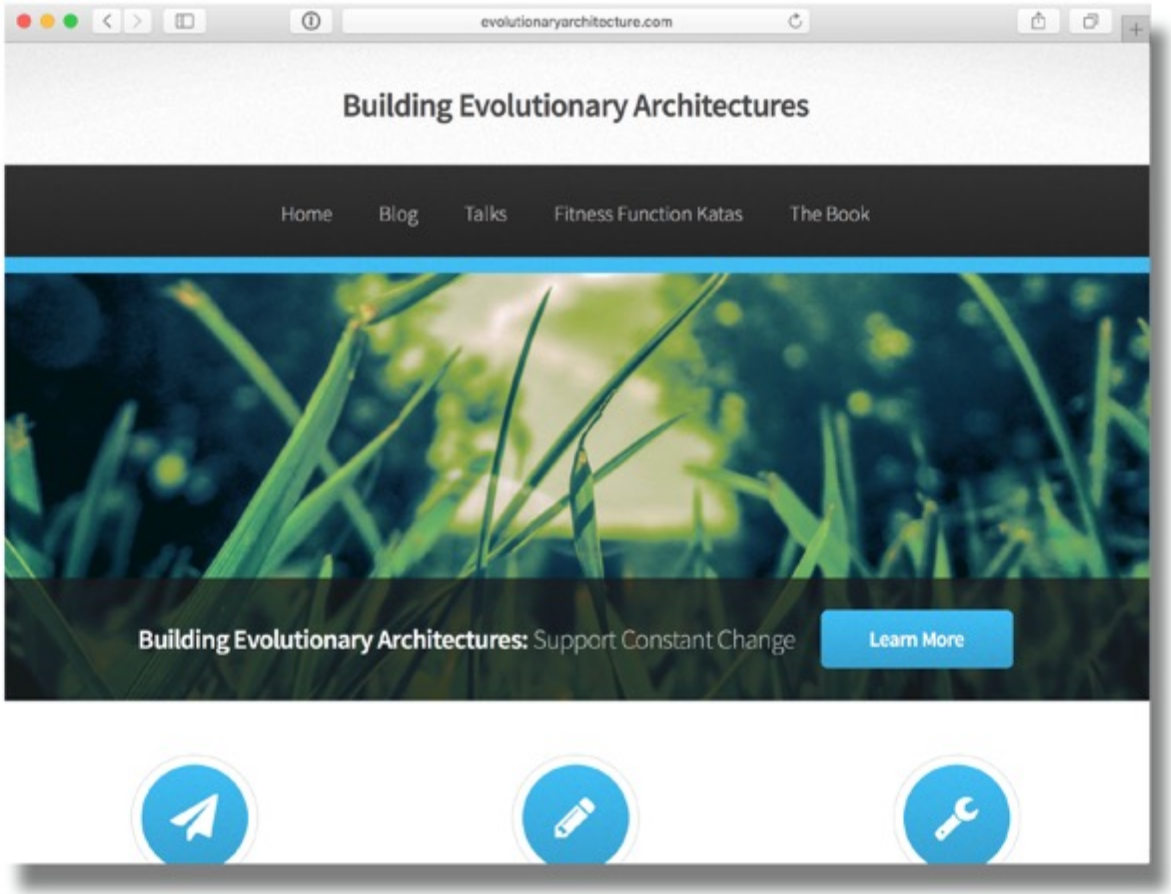
prioritization

Architecture Defined

architecture =  + 

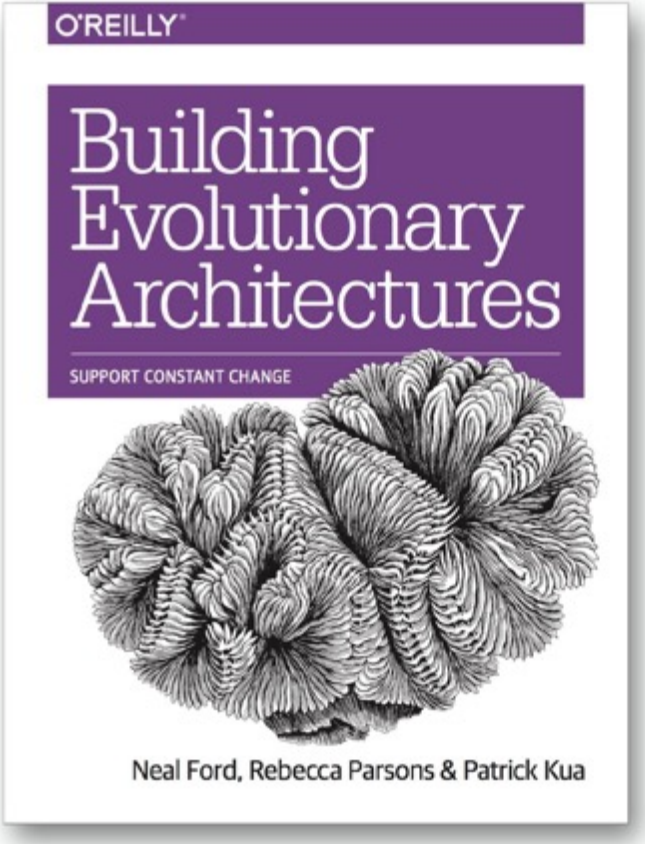
prioritization

cost



http://evolutionaryarchitecture.com

iMpLemENtiNg eVoLuTiONaRy ARcHitECtuREs



@neal4d
nealford.com

@rebeccaparsons

@patkua