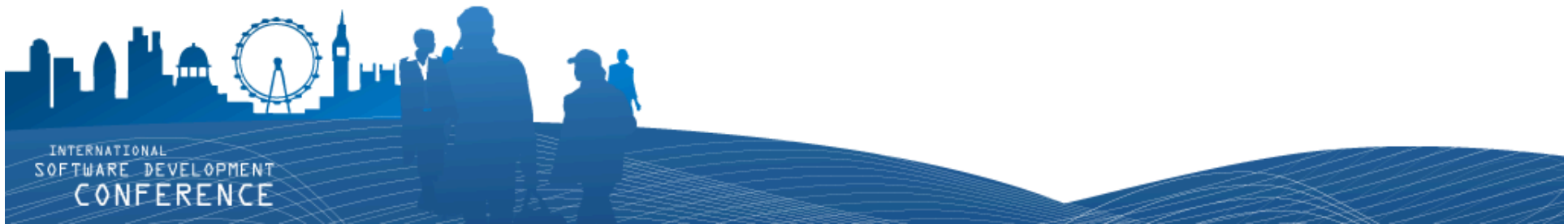
QCon

Please evaluate my talk via the mobile app!



enterprise REST
a case study

BRANDON BYARS principal consultant

ThoughtWorks®

bbyars@thoughtworks.com

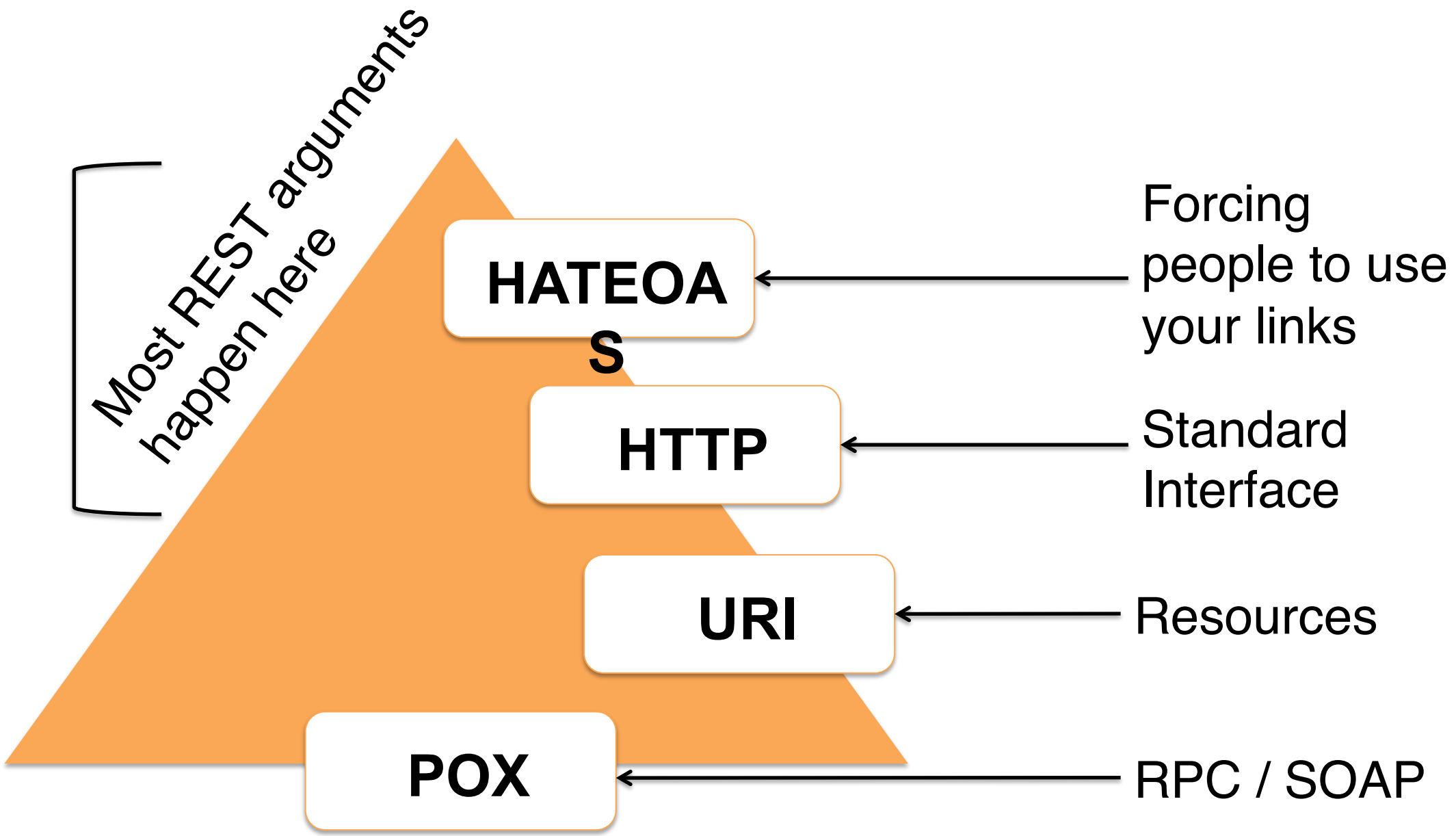
Suite 600, 15455 Dallas Parkway , Addison, TX 75001

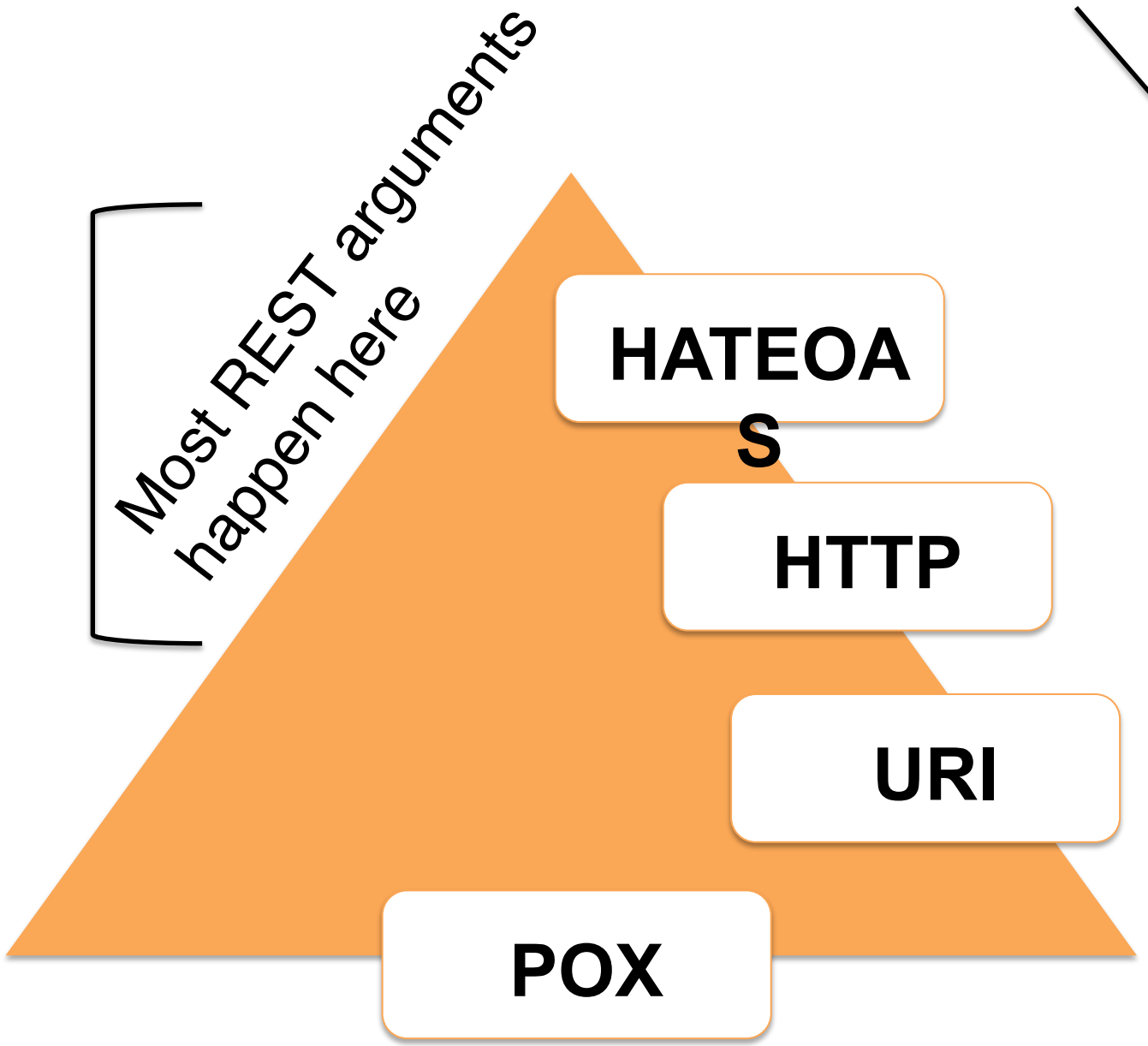
tel 415.722.1343

twitter:@BrandonByars

the eight fallacies of distributed programming

- the network is reliable
- latency is zero
- bandwidth is infinite
- the network is secure
- topology doesn’t change
- there is one administrator
- transport cost is zero
- the network is homogenous


Most REST arguments happen here
HATEOAS
HTTP
URI
POX
Forcing people to use your links
Standard Interface
Resources
RPC / SOAP



Most REST mistakes happen over here…

versioning

deployment

testing

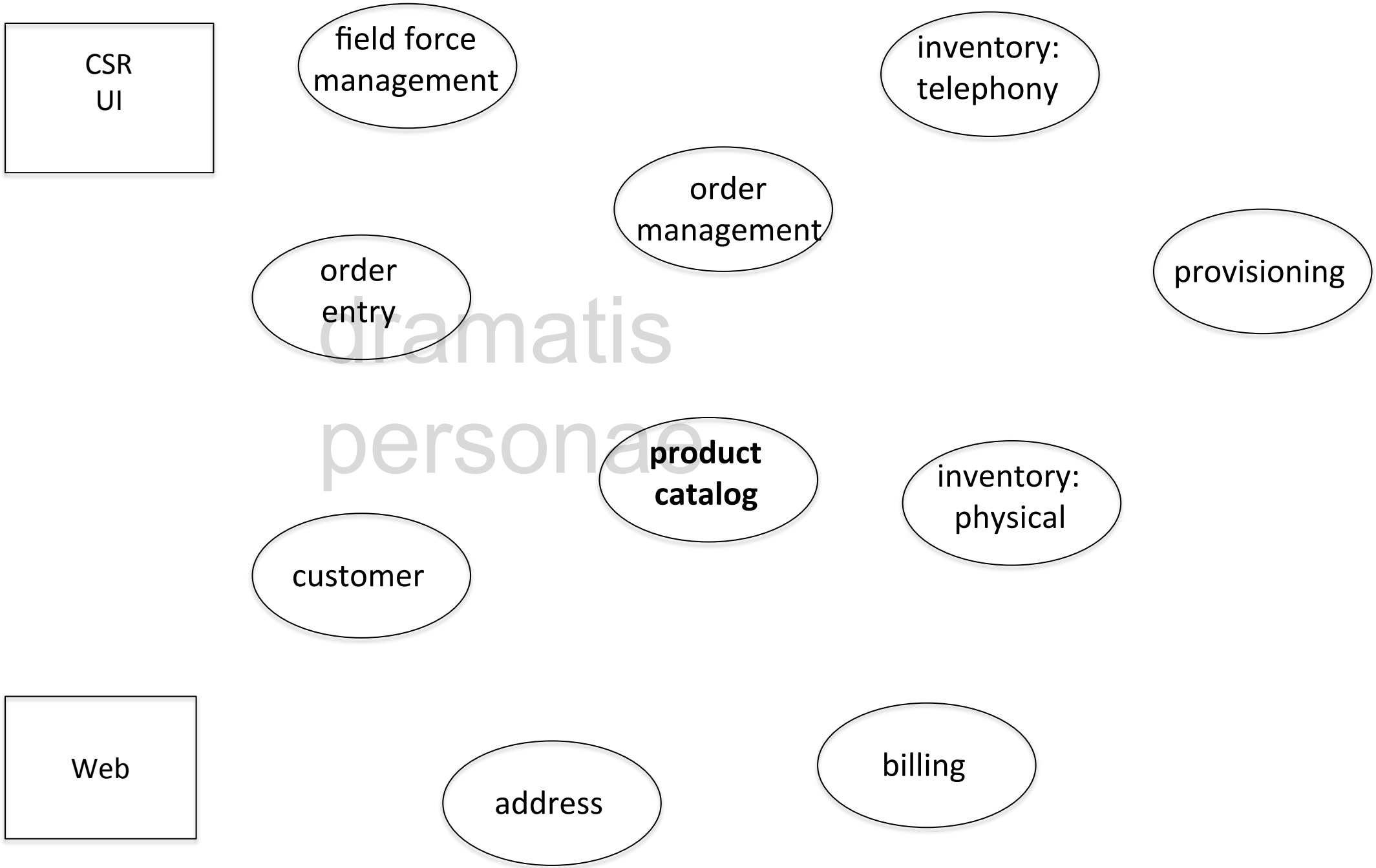
service granularity


ENTERPRISE
REST

a story about a billing system...

a story about a legacy rewrite...

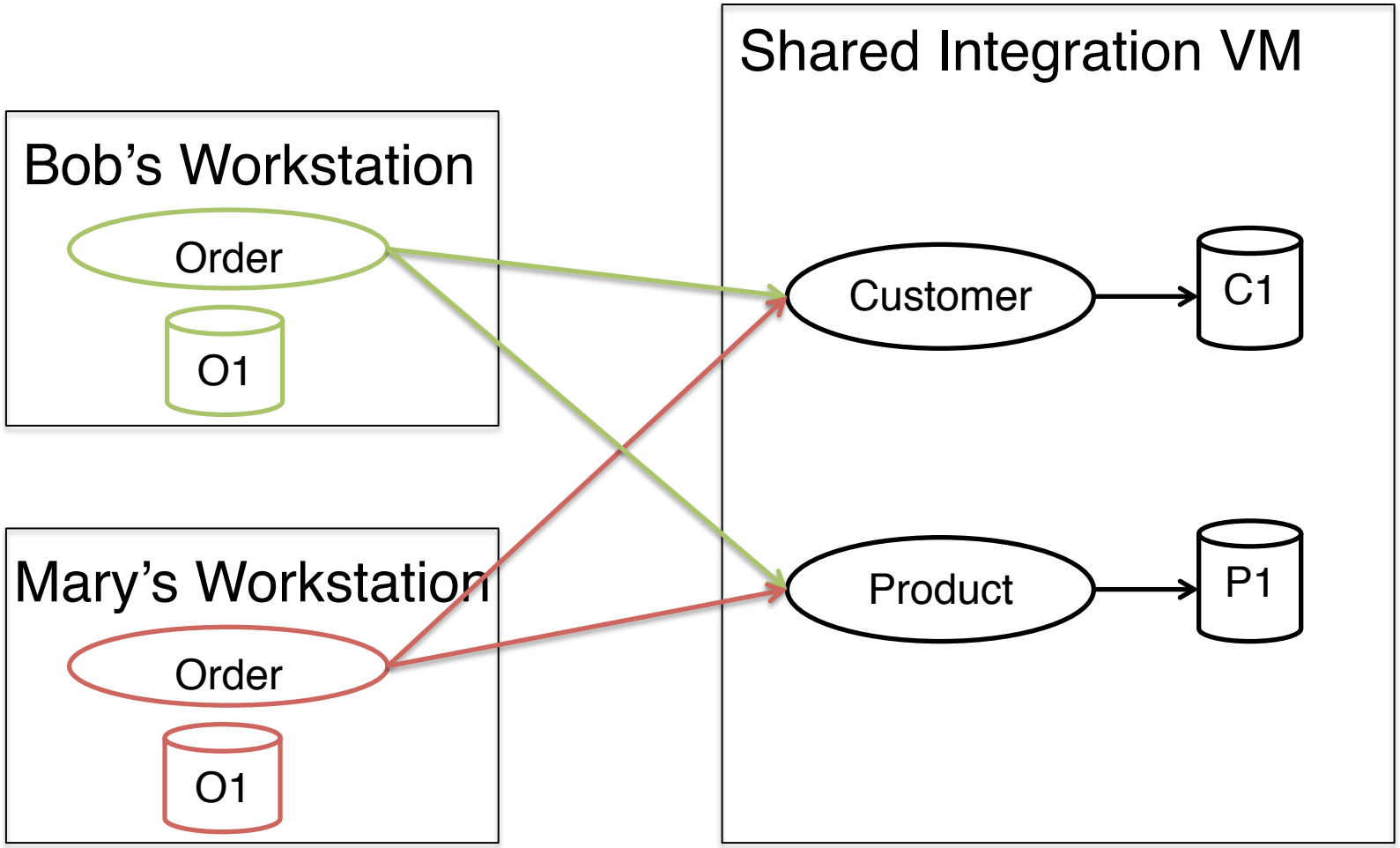
dramatis
personae

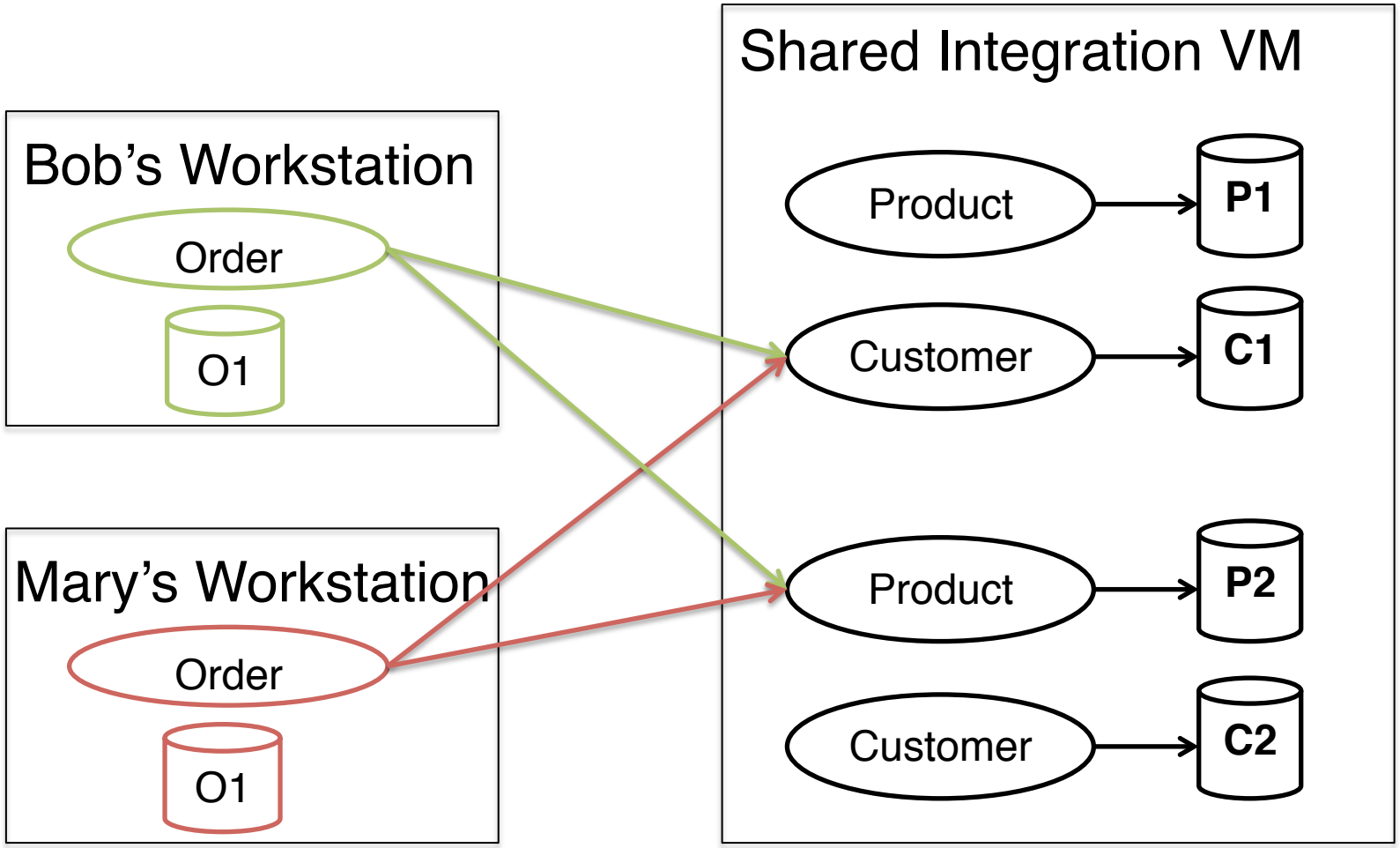

CSR
UI
field force
management
inventory:
telephony
order
management
order
entry
provisioning
dramatis
personae
product
catalog
inventory:
physical
customer
Web
address
billing

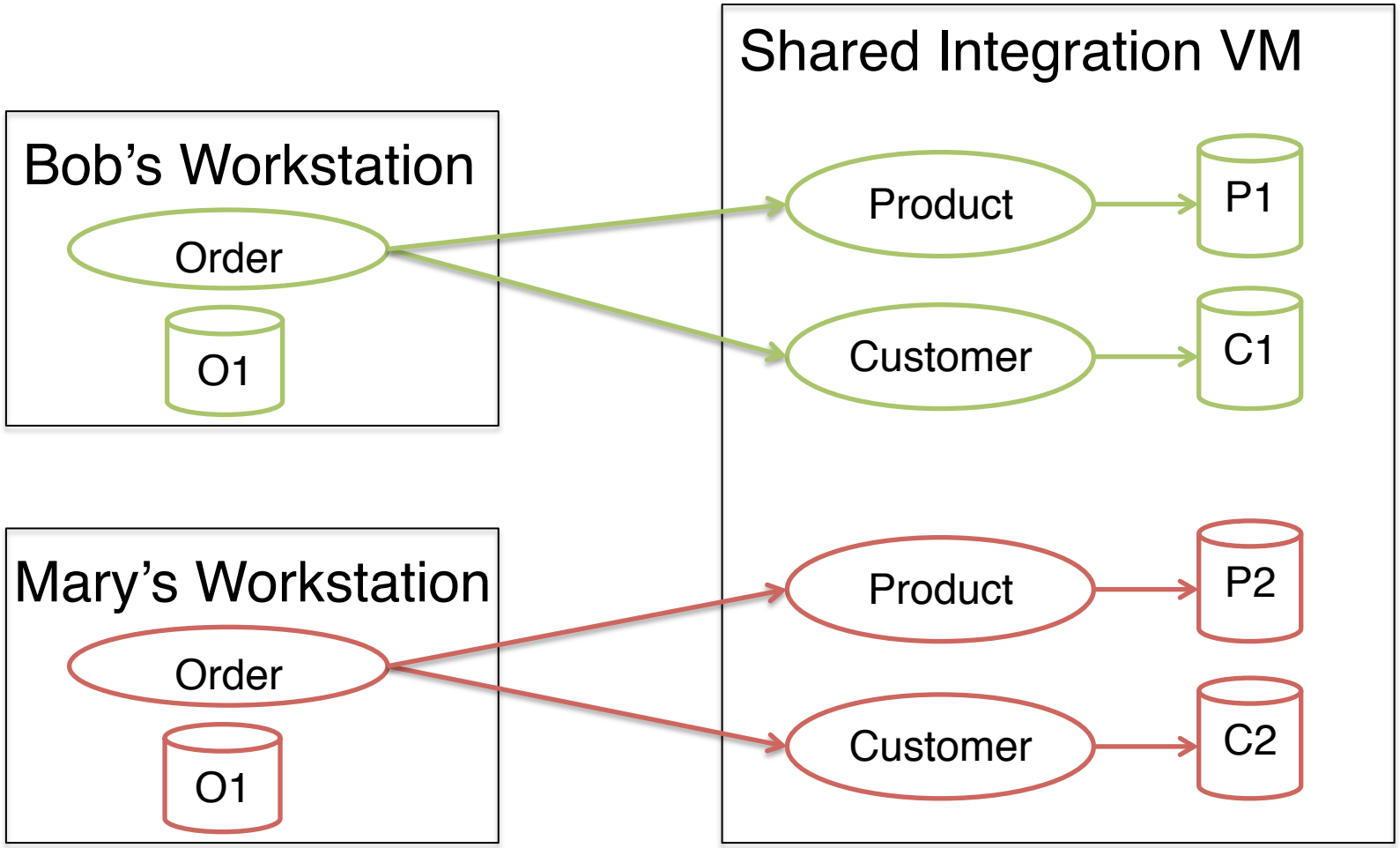
Choreography...

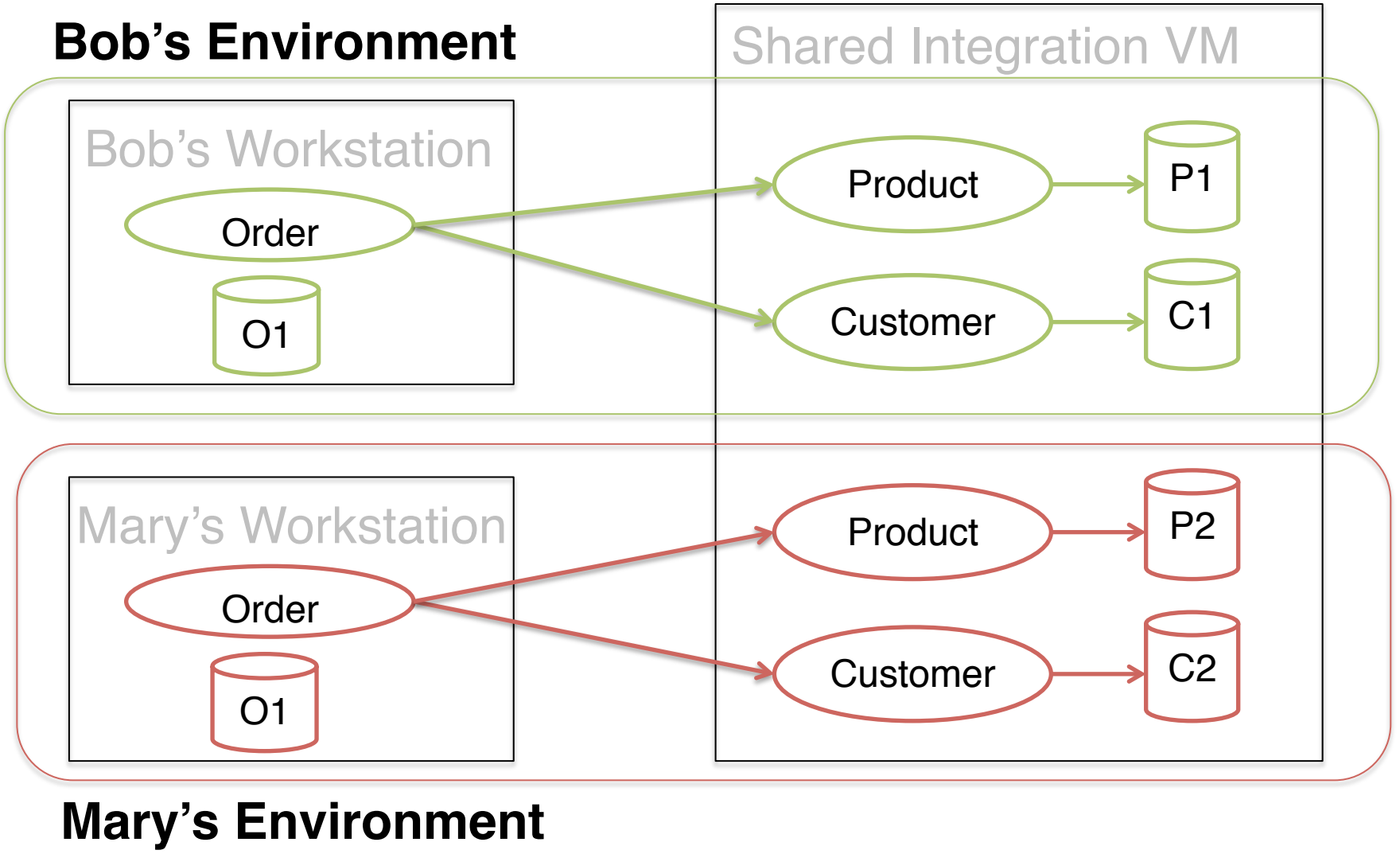
...not orchestrati

define logical environments for isolation


Shared Integration VM
Bob's Workstation
Order
O1
Mary's Workstation
Order
O1
Customer
C1
Product
P1


Shared Integration VM
Bob's Workstation
Order
O1
Mary's Workstation
Order
O1
Product
P1
Customer
C1
Product
P2
Customer
C2


Shared Integration VM
Bob's Workstation
Order
O1
Product
P1
Customer
C1
Mary's Workstation
Order
O1
Product
P2
Customer
C2


Bob's Environment
Shared Integration VM
Bob's Workstation
Order
O1
Product
P1
Customer
C1
Mary's Workstation
Order
O1
Product
P2
Customer
C2
Mary's Environment

coordinated deployments

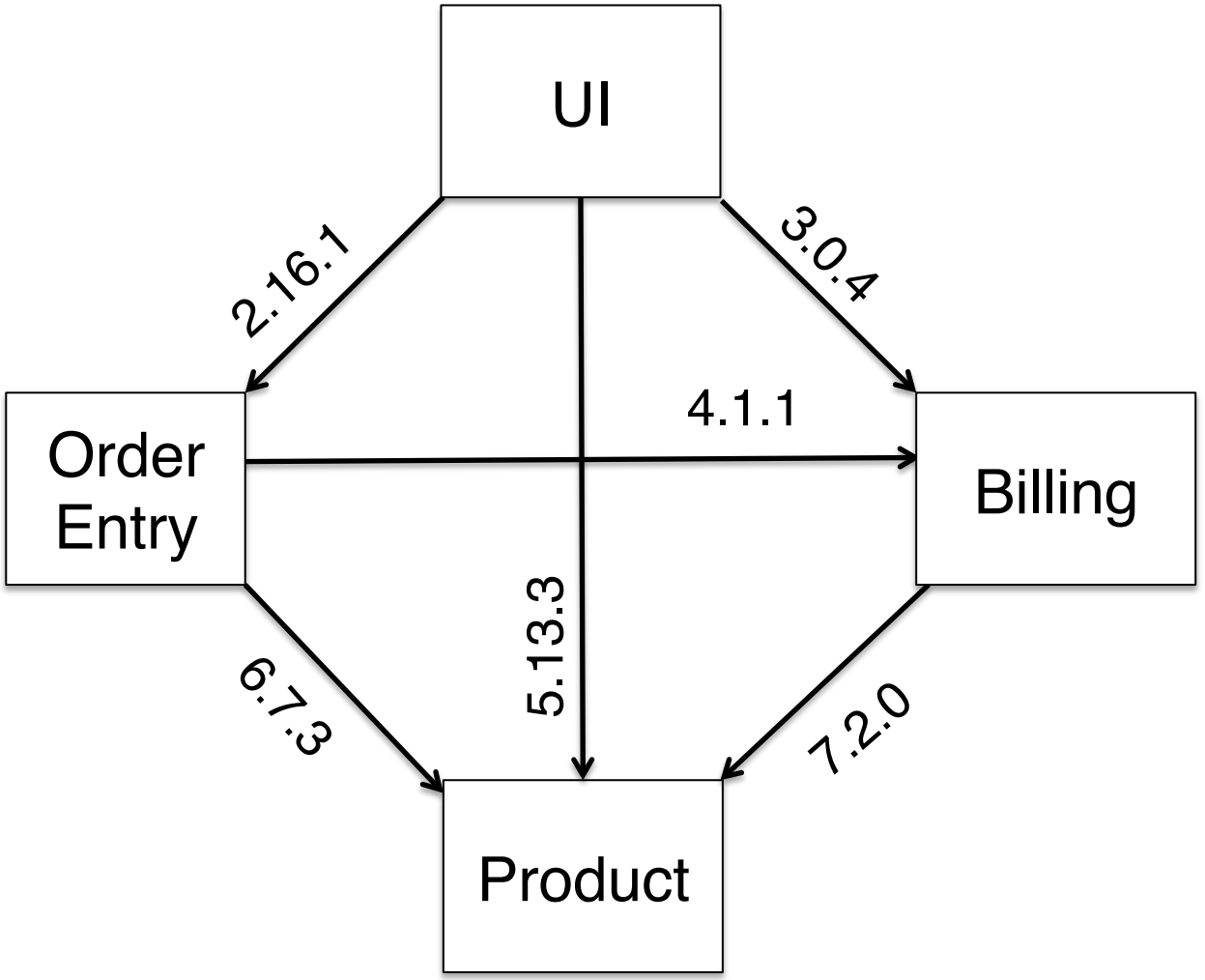
```
order-bob:
  product:
    webServers: [s1.test.dev]
    url: http://s1.test.dev:8000/products
    dbServer: s1.test.dev
    dbName: OrderBobProducts
  customer:
    webServers: [s1.test.dev]
    url: http://s1.test.dev:8001/customers
    dbServer: s1.test.dev
    dbName: OrderBobCustomers
```

```
order-mary:
  product:
    webServers: [s1.test.dev]
```

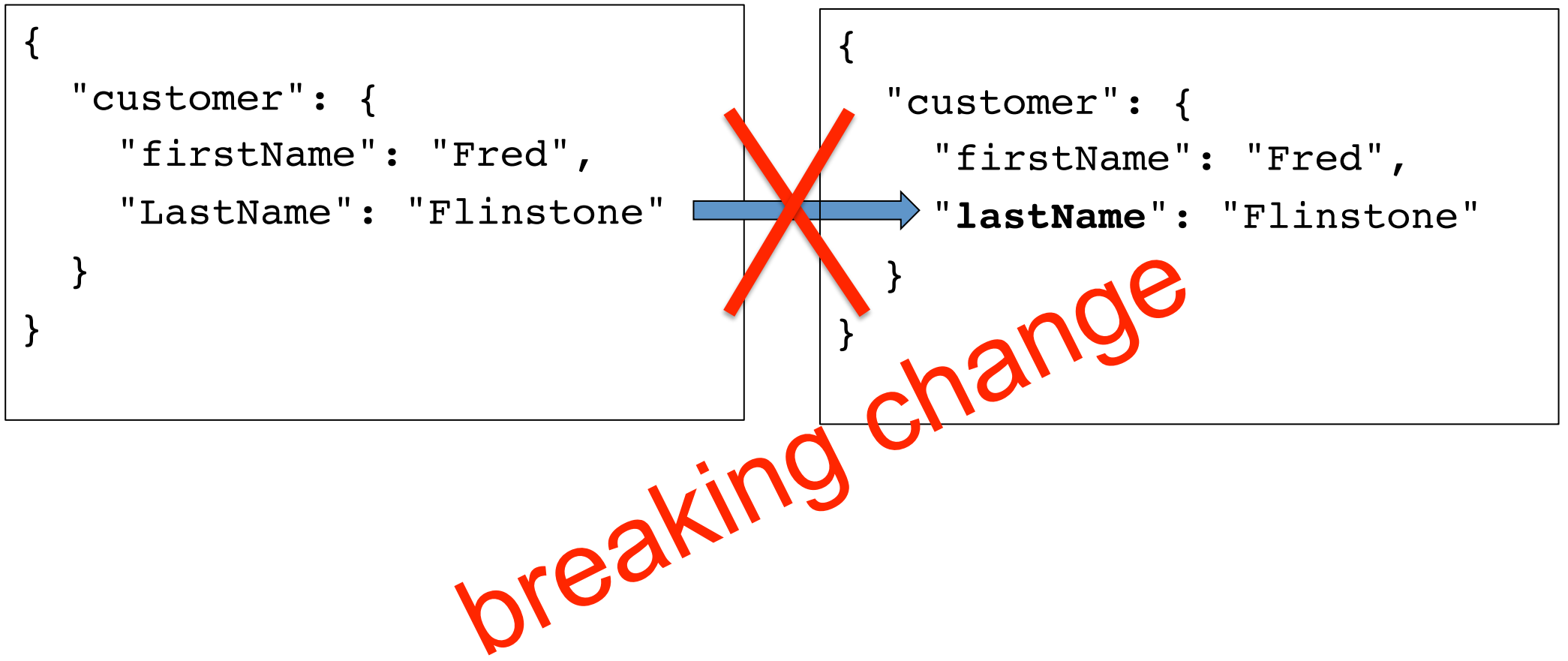
```
order-bob:
  product:
    webServers: [s1.test.dev]
    url: http://s1.test.dev:8000/products
    dbServer: s1.test.dev
    dbName: OrderBobProducts
  customer:
    webServers: [s1.test.dev]
    url: http://s1.test.dev:8001/customers
    dbServer: s1.test.dev
    dbName: OrderBobCustomers

order-mary:
  product:
    webServers: [s1.test.dev]
```

version as a last resort


UI
2.16.1
3.0.4
4.1.1
Order Entry
Billing
5.13.3
6.7.3
7.2.0
Product

speling mistaeks



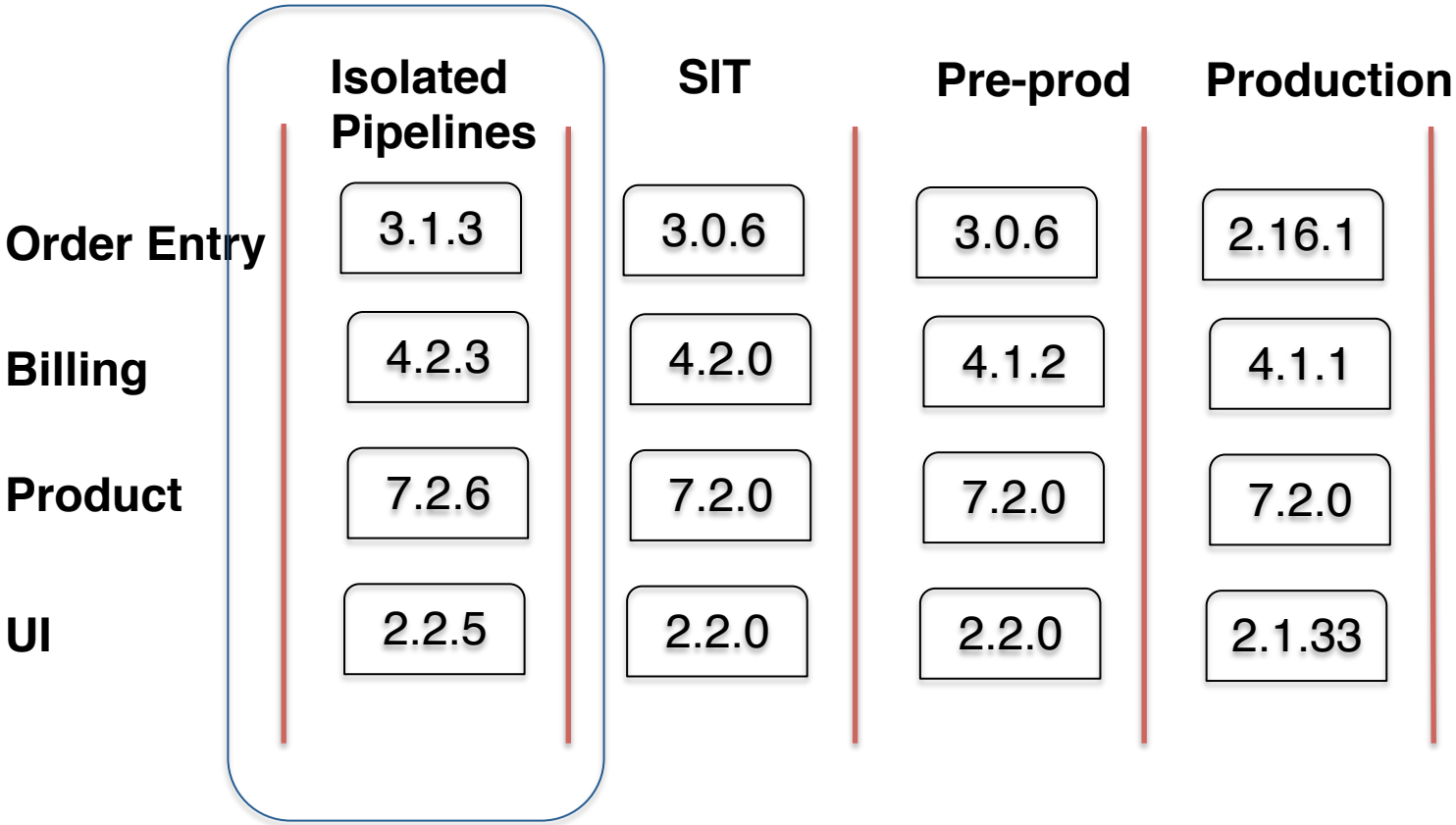
Postel's Law

```
@RequestMapping(method=POST, produces=application/json)
@ResponseBody
public OrderResource createOrder(
    @RequestBody Order order,
    HttpServletResponse response
) throws OrderException
{
    ...
    ...
}
```

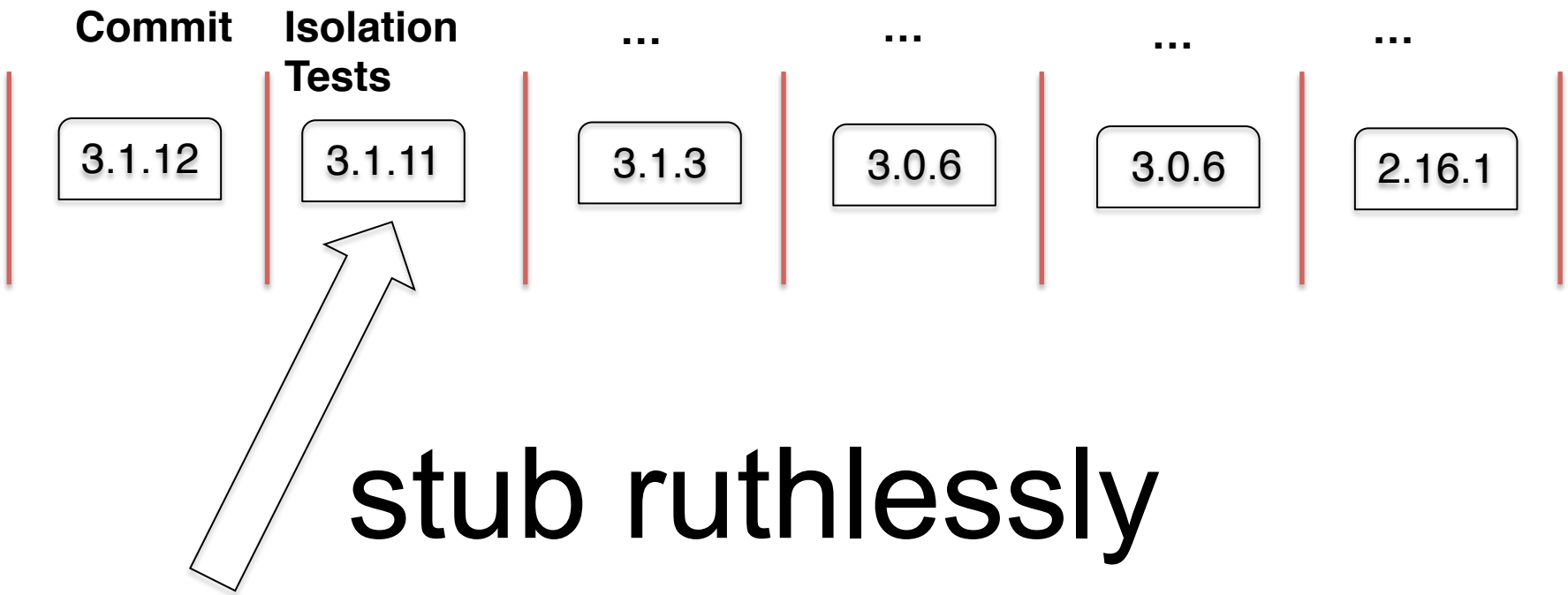
separate functional testing from integration testing

deployment pipeline

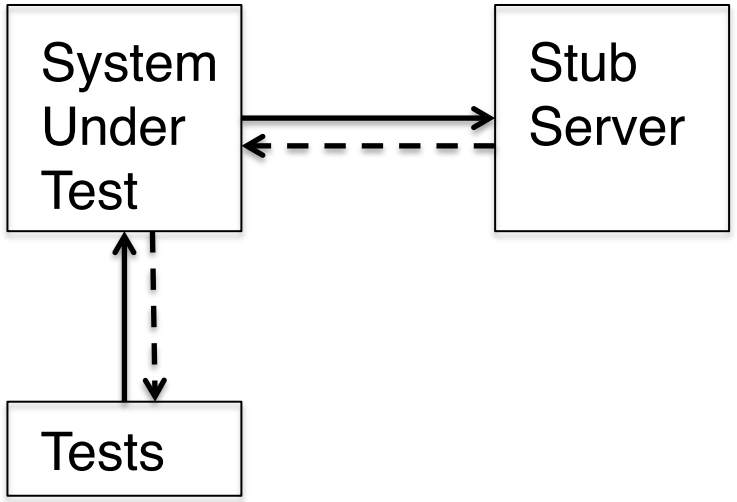
deployment pipelines...



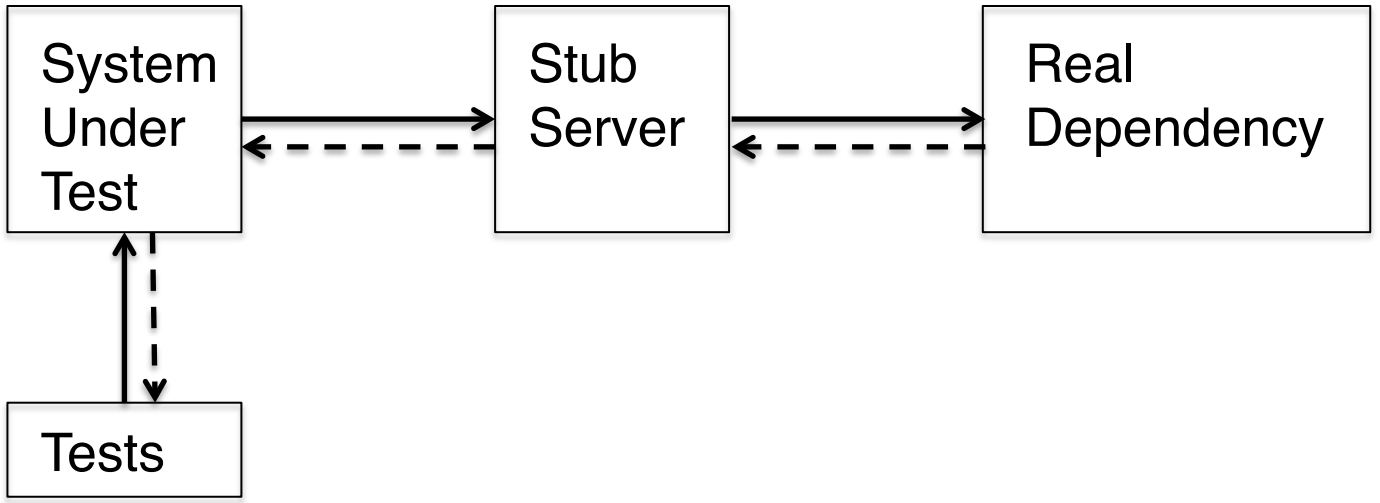
early stages



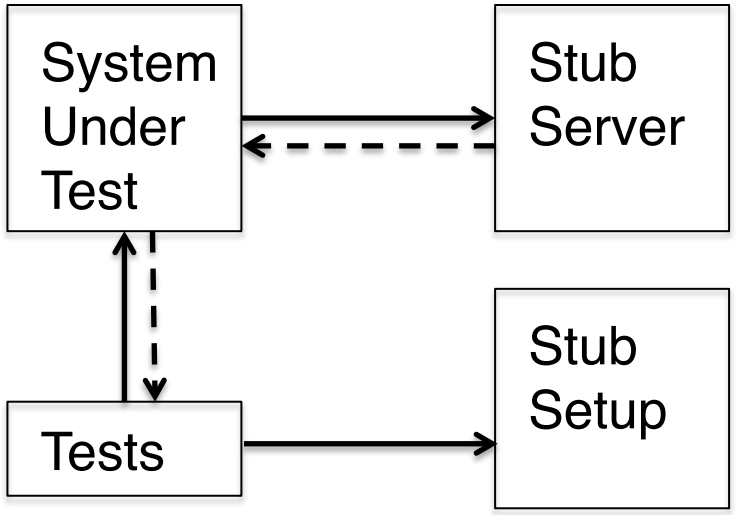
hand-crafted stubs



record and replay stubs



general purpose stubs



Moco: https://github.com/dreamhead/moco

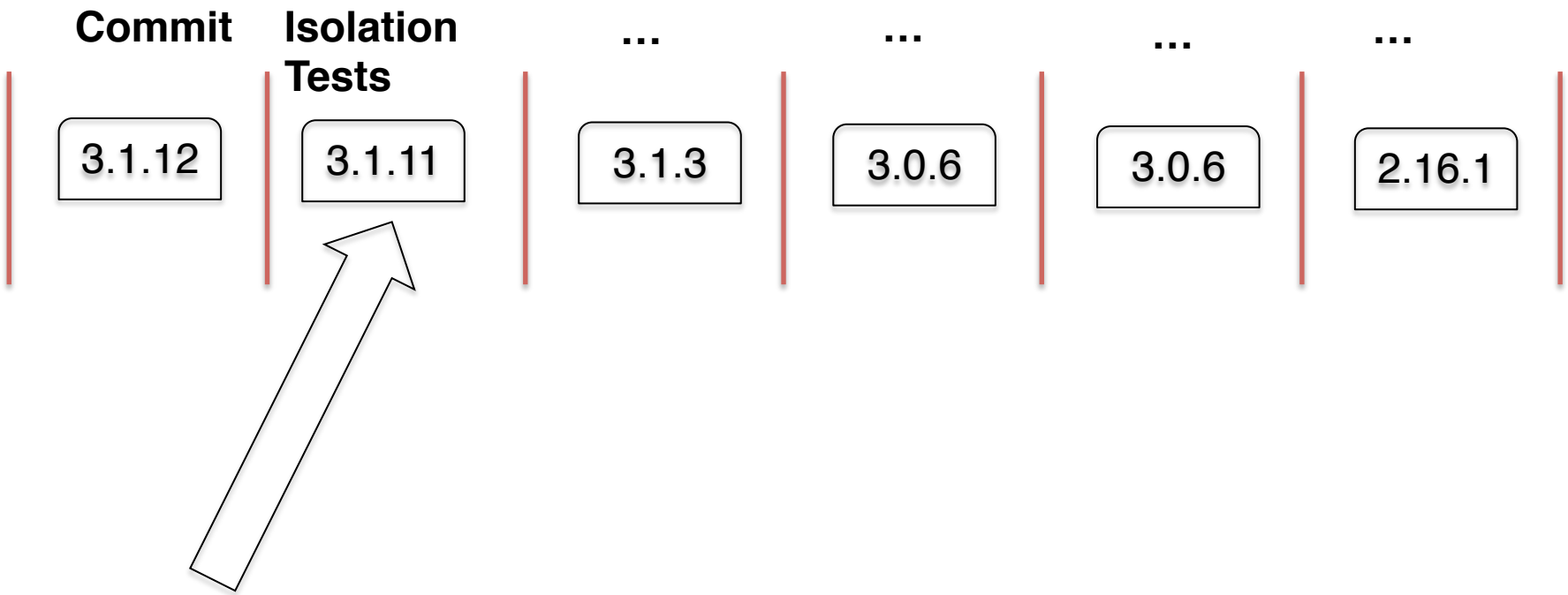
vcr: https://github.com/vcr/vcr

βetamax: https://github.com/robfletcher/betamax

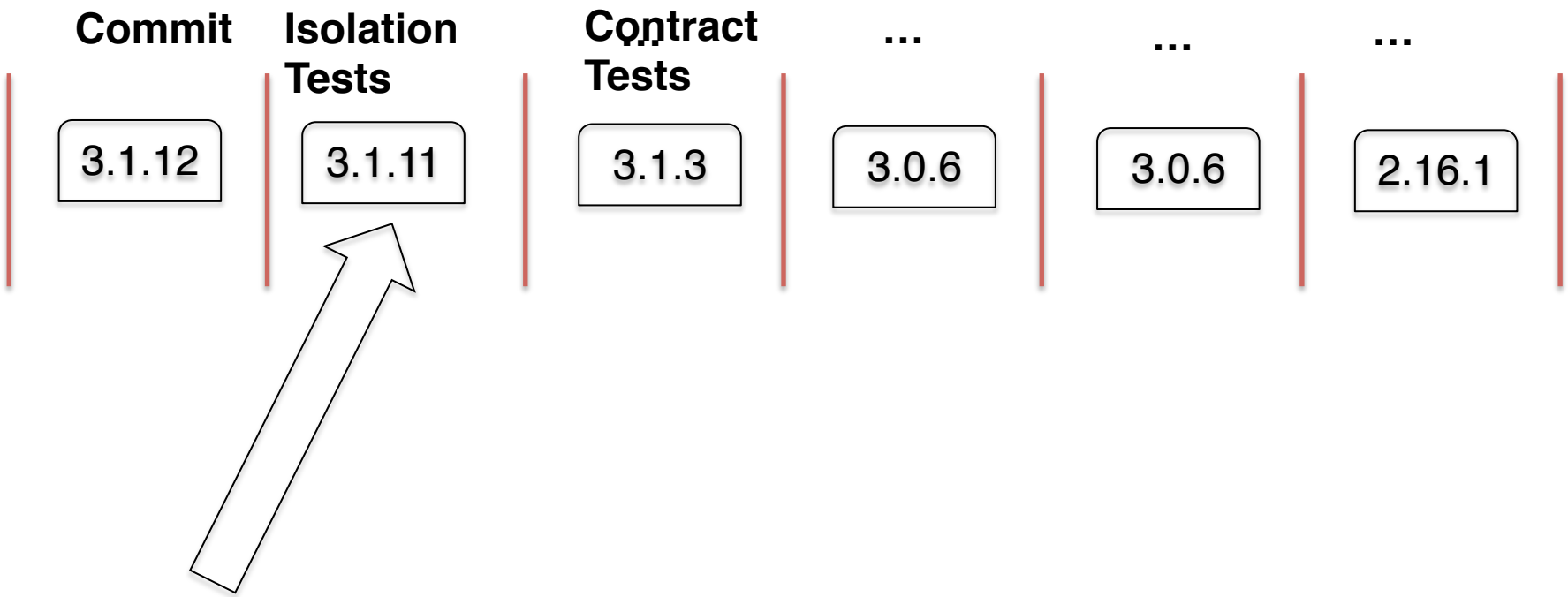
stubby4j: https://github.com/azagniotov/stubby4j

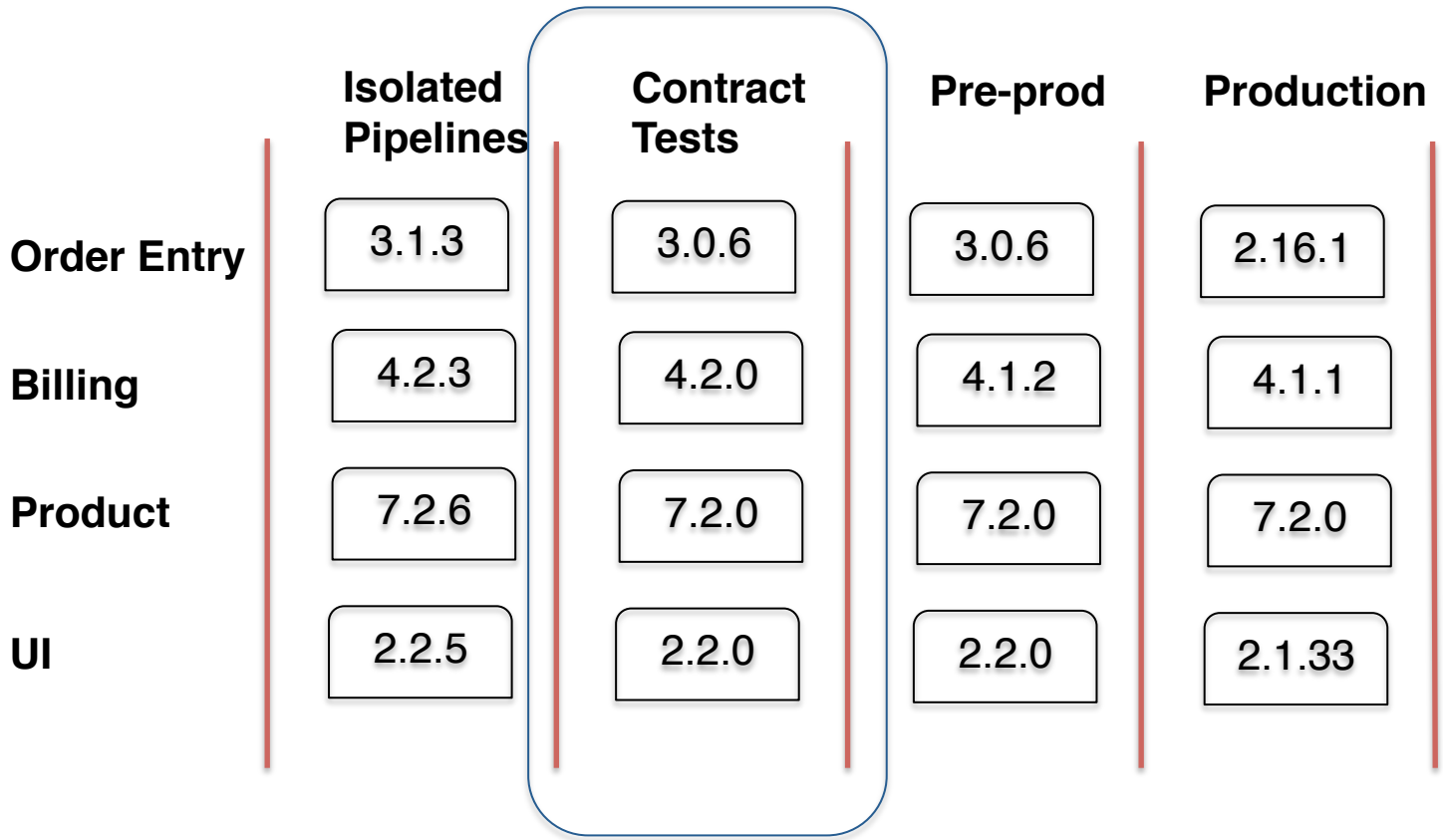
mountebank: http://www.mbtest.org/

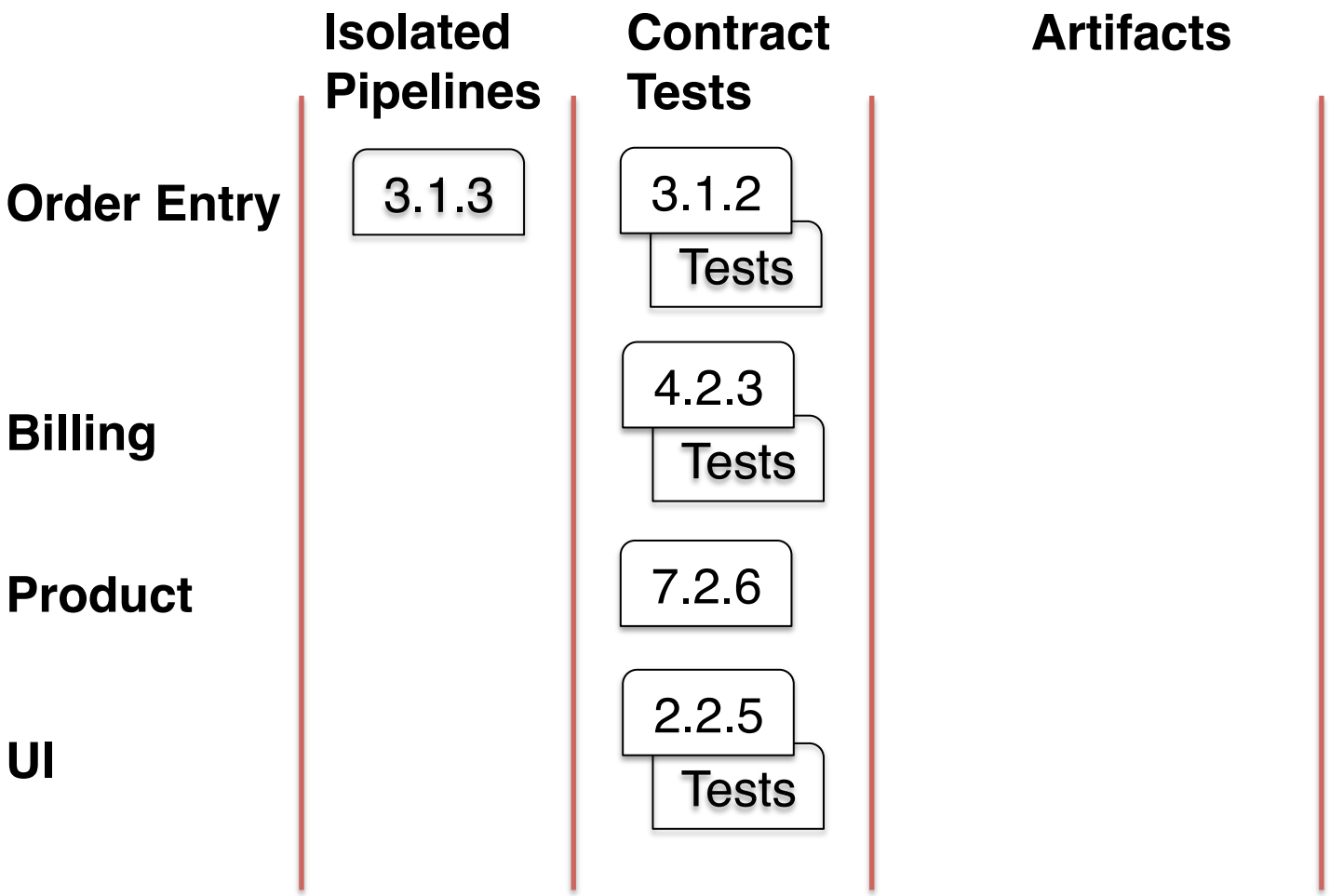
early stages

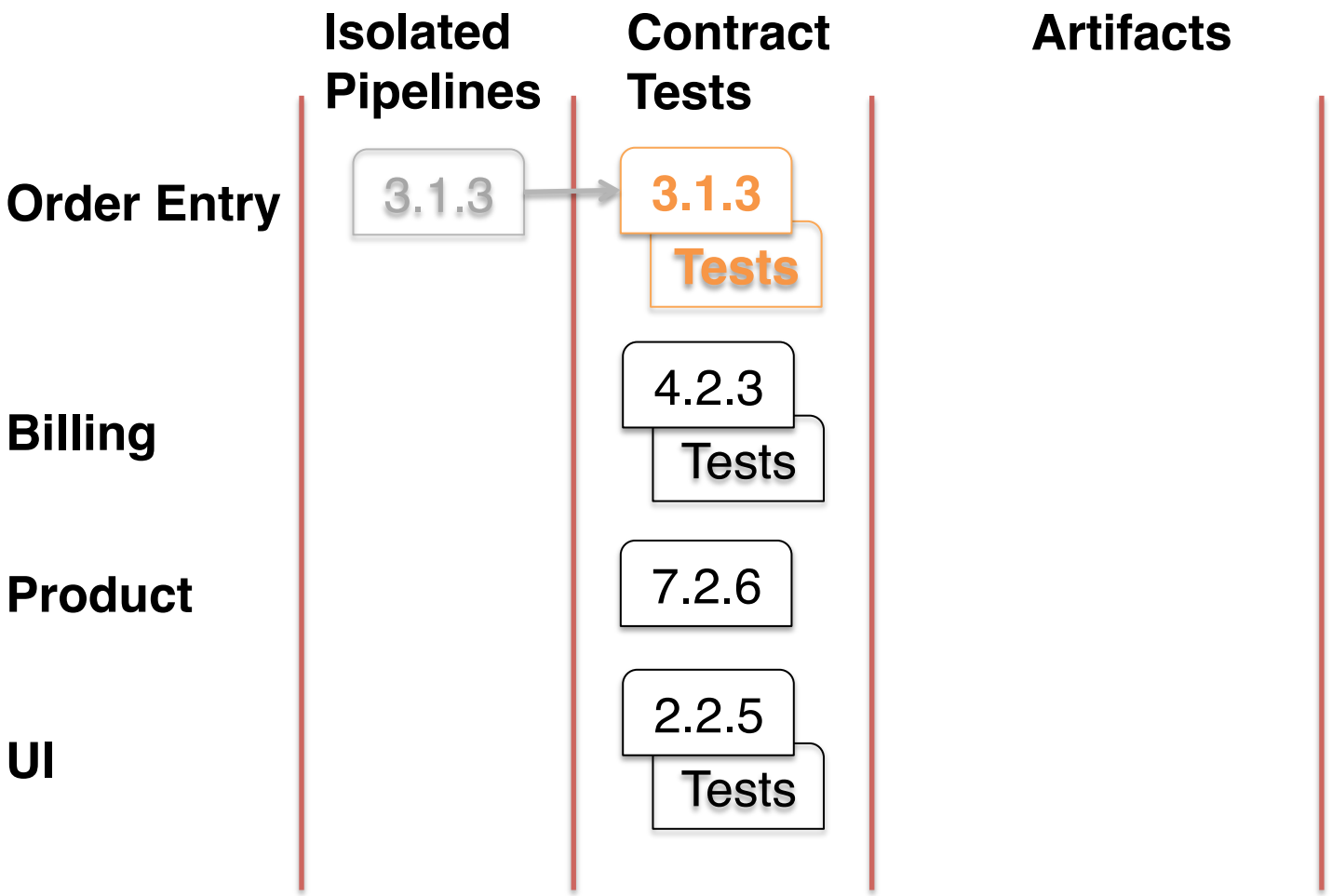


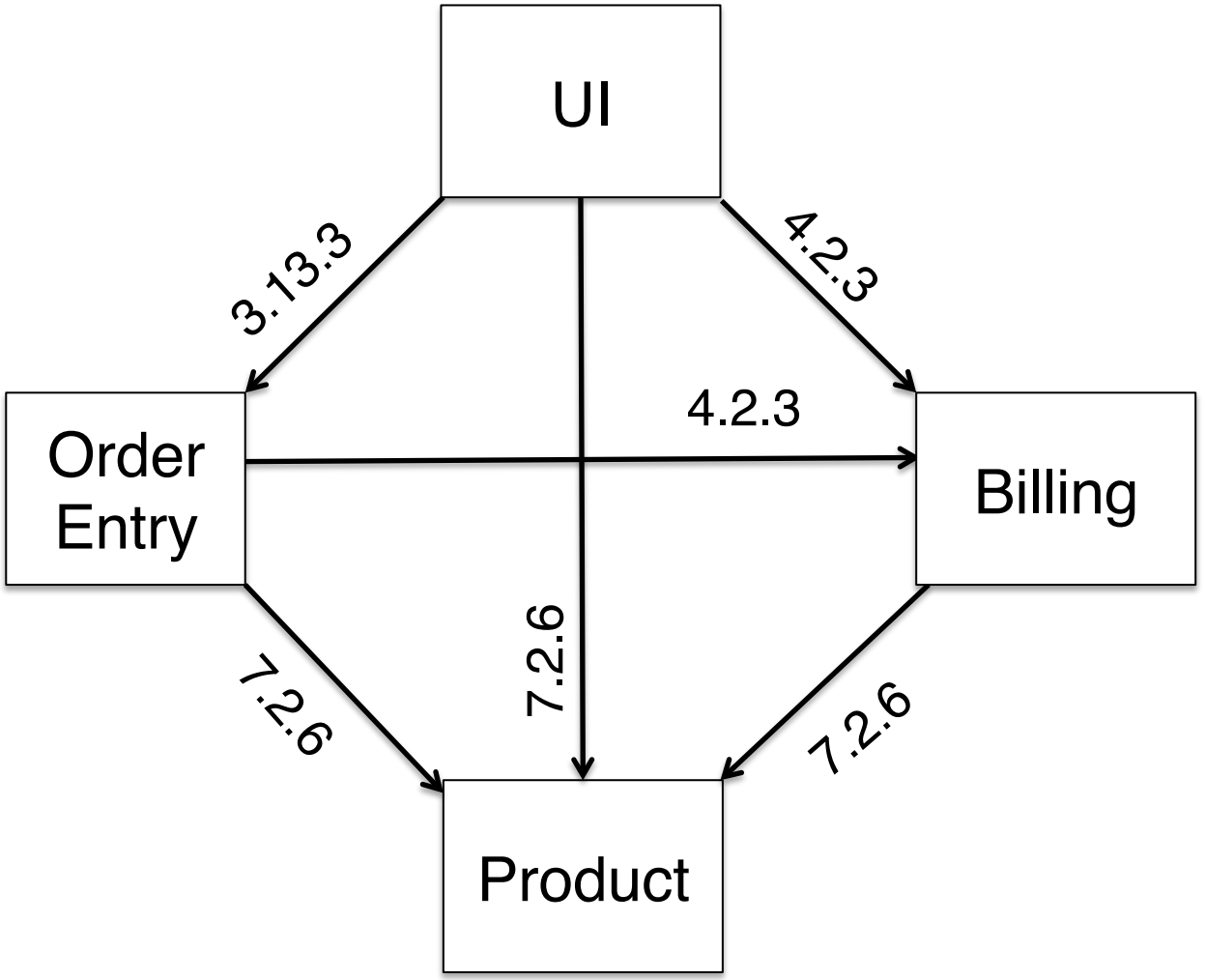
early stages

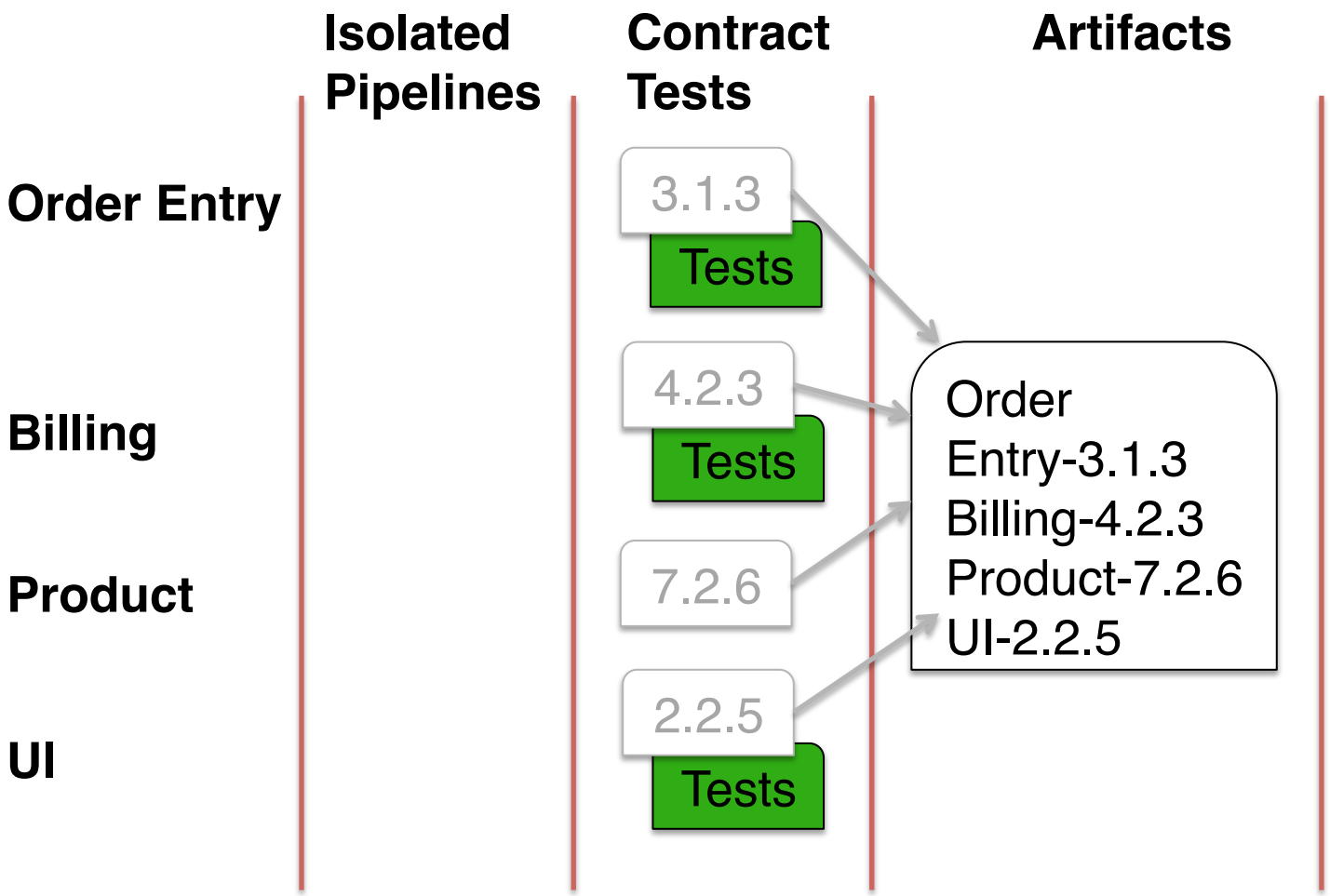


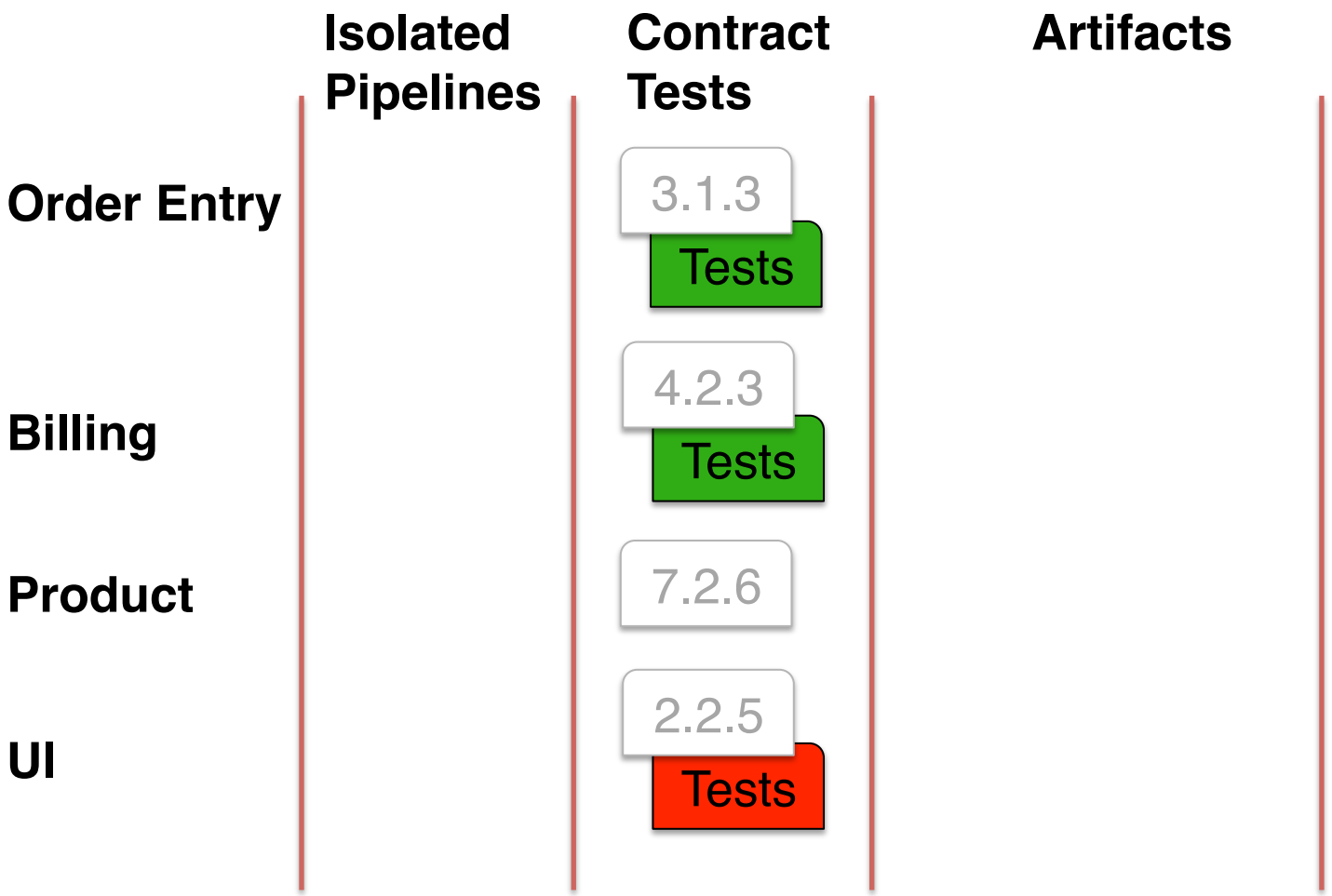

Isolated Pipelines
Contract Tests
Pre-prod
Production
Order Entry
3.1.3
3.0.6
3.0.6
2.16.1
Billing
4.2.3
4.2.0
4.1.2
4.1.1
Product
7.2.6
7.2.0
7.2.0
7.2.0
UI
2.2.5
2.2.0
2.2.0
2.1.33


Isolated Pipelines
Contract Tests
Artifacts
Order Entry
3.1.3
3.1.2
Tests
Billing
4.2.3
Tests
Product
7.2.6
UI
2.2.5
Tests


Isolated Pipelines
Contract Tests
Artifacts
Order Entry
3.1.3
3.1.3
Tests
Billing
4.2.3
Tests
Product
7.2.6
UI
2.2.5
Tests


UI
3.13.3
4.2.3
Order Entry
4.2.3
Billing
7.2.6
7.2.6
7.2.6
Product

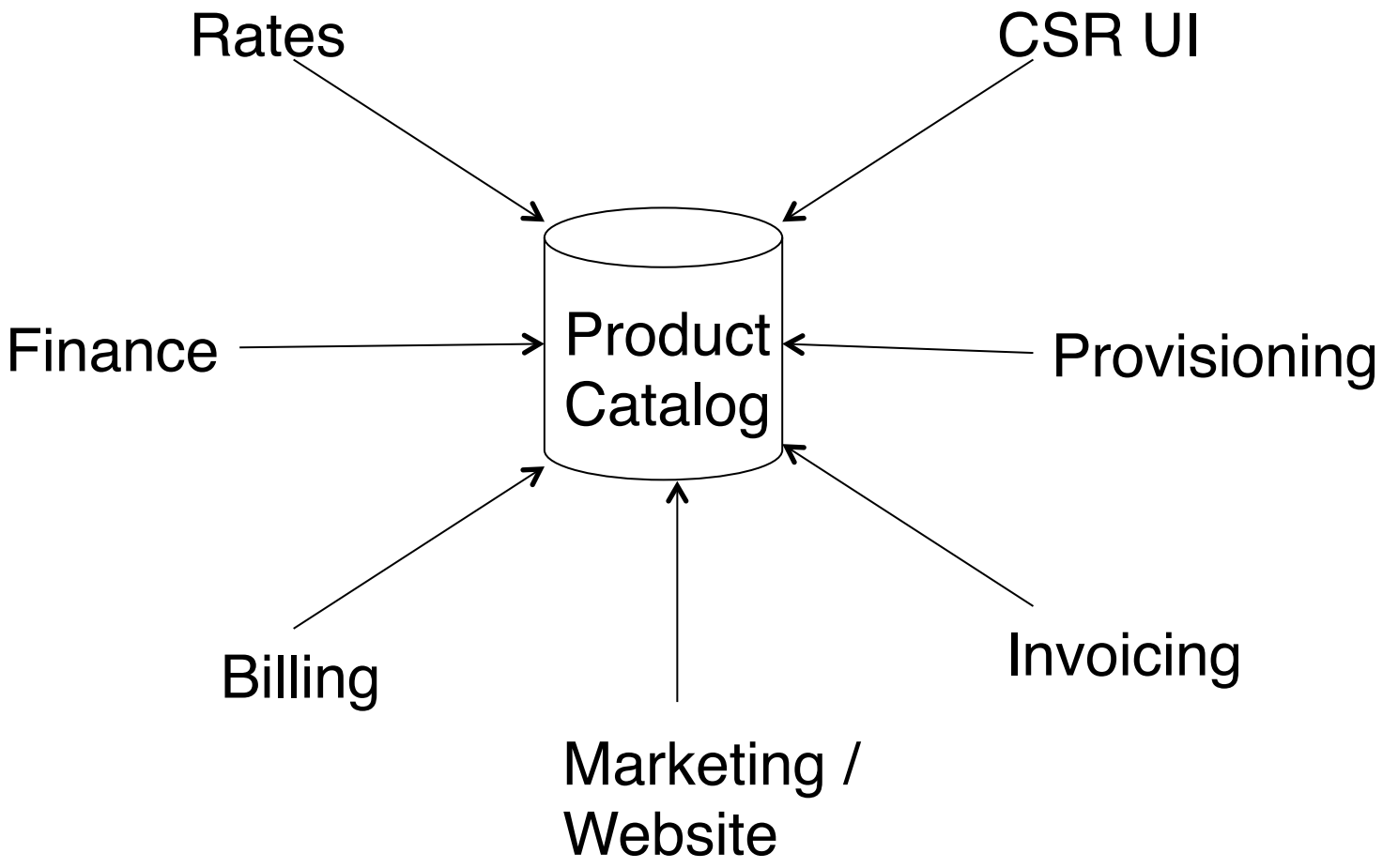

Isolated Pipelines
Contract Tests
Artifacts
Order Entry
Billing
Product
UI
3.1.3
Tests
4.2.3
Tests
7.2.6
2.2.5
Tests
Order Entry-3.1.3
Billing-4.2.3
Product-7.2.6
UI-2.2.5


Isolated Pipelines
Contract Tests
Artifacts
Order Entry
3.1.3
Tests
Billing
4.2.3
Tests
Product
7.2.6
UI
2.2.5
Tests

```
[Test]
public void ValidateProductAttributes()
{
    var url = UrlForTestProduct();
    var response = new HttpResource(url)
                        .ThatAccepts("application/xml")
                        .Get();

    Assert.That(response.StatusCode, Is.EqualTo(200));
    AssertHasXPath(response.Body, "//productCode");
    AssertHasXPath(response.Body, "//description");
    AssertHasXPath(response.Body, "//monthlyCharge");
    AssertNumeric(ValueFor(response.Body, "//monthlyCharge"));
}
```

use bounded contexts
to control complexity

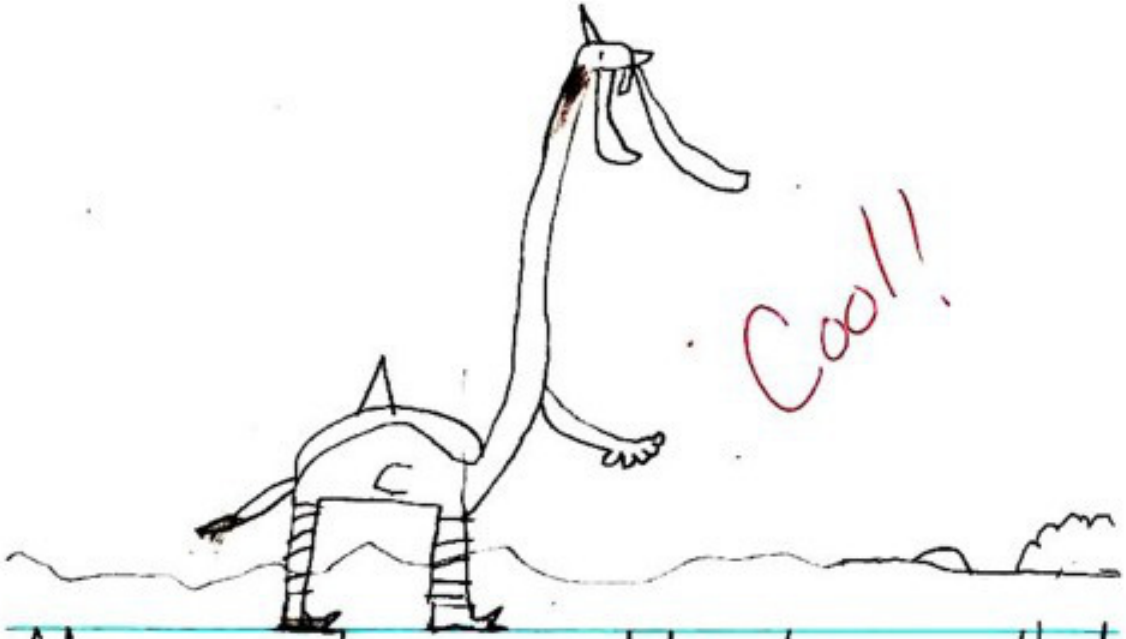

Rates
CSR UI
Finance
Product
Catalog
Provisioning
Billing
Marketing /
Website
Invoicing

data rationalization

Conway's Law

Organizations which design systems are constrained to produce designs which are copies of the communication structures of these organizations

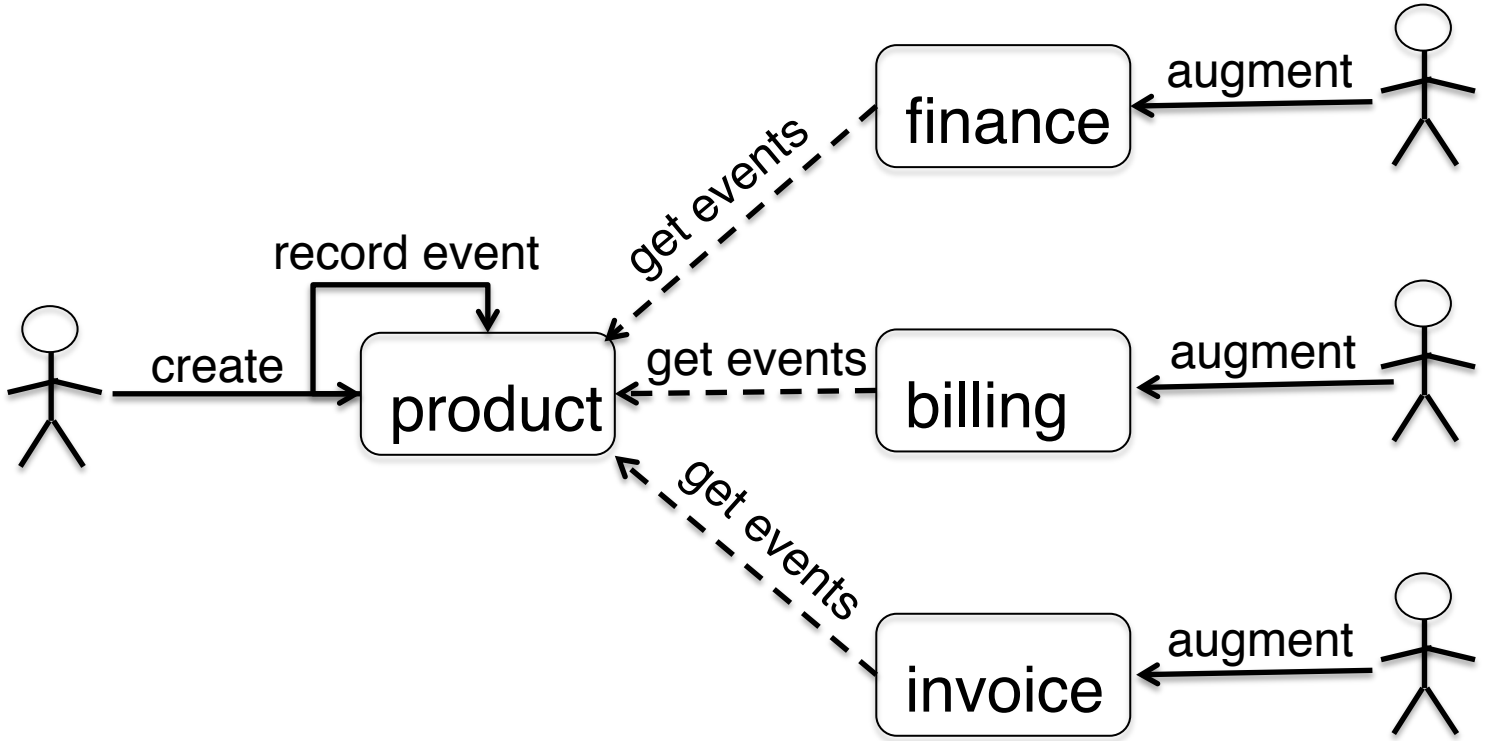
In every organization there will always be one person who knows what is going on. This person must be hired.



My pet would be called Nugger-fugger. Human, shark, elephant, bird, bear, beaver, zebra, wasp, cat and hyeana. He would be an anphibean.

By Jackson Byars. 8-20-06

contextualize!



BRANDON BYARS principal consultant

ThoughtWorks®

bbyars@thoughtworks.com

Suite 600, 15455 Dallas Parkway , Addison, TX 75001

tel 415.722.1343

twitter:@BrandonByars

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

QCon

Please evaluate my talk via the mobile app!