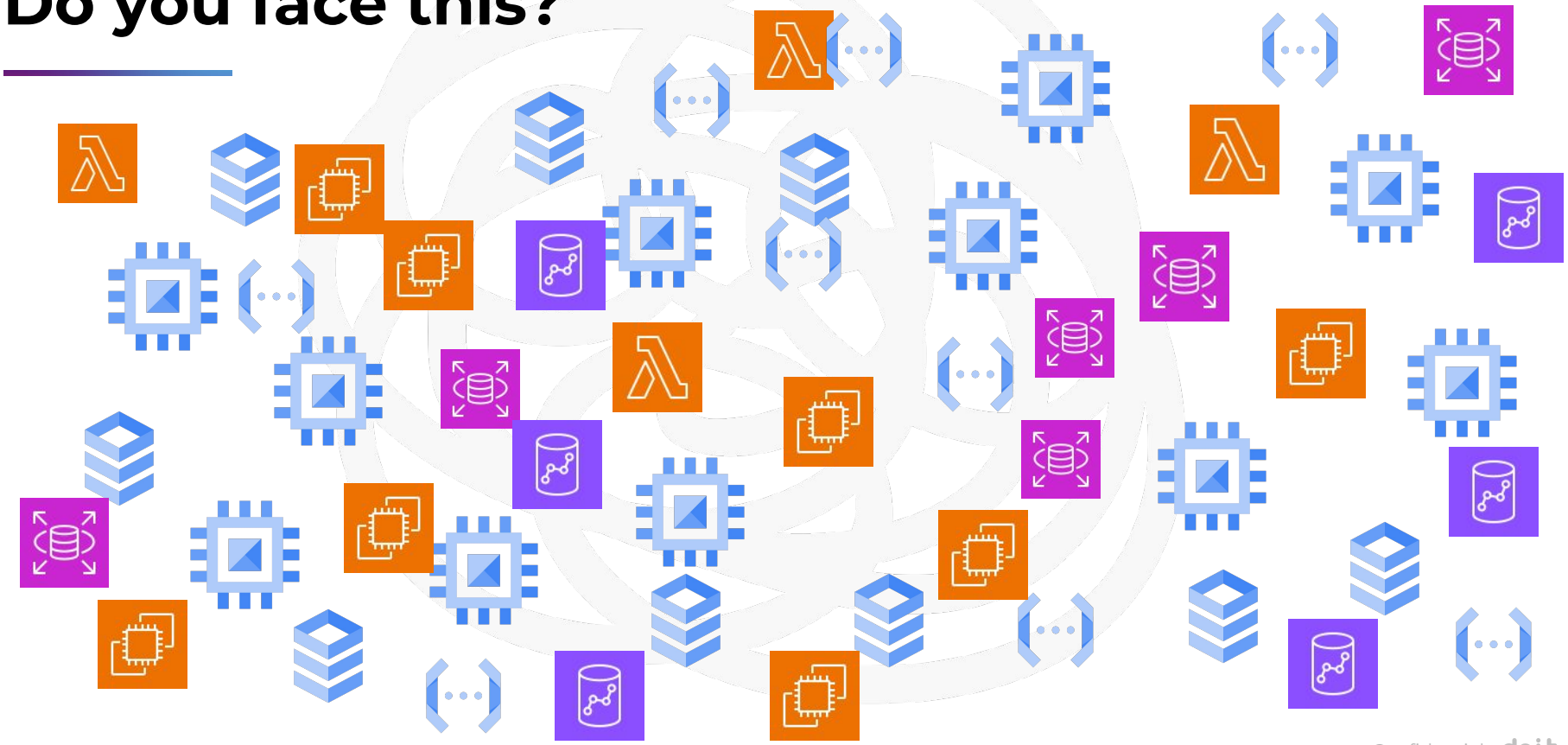
Untangling the Tangled Cloud

Joshua Fox
Senior Cloud Architect

Joshua Fox

joshua@doit-intl.com
Senior Cloud Architect
DoiT

Do you face this?



What you will learn

How to sort out your cloud resources, in whatever cloud

How we will do it

We will see what a good dependency graph looks like:

The boundaries should enable that.

What you will gain

Seven mental models for organizing the cloud,
based on 20 years architecture experience.

Technology makes it real

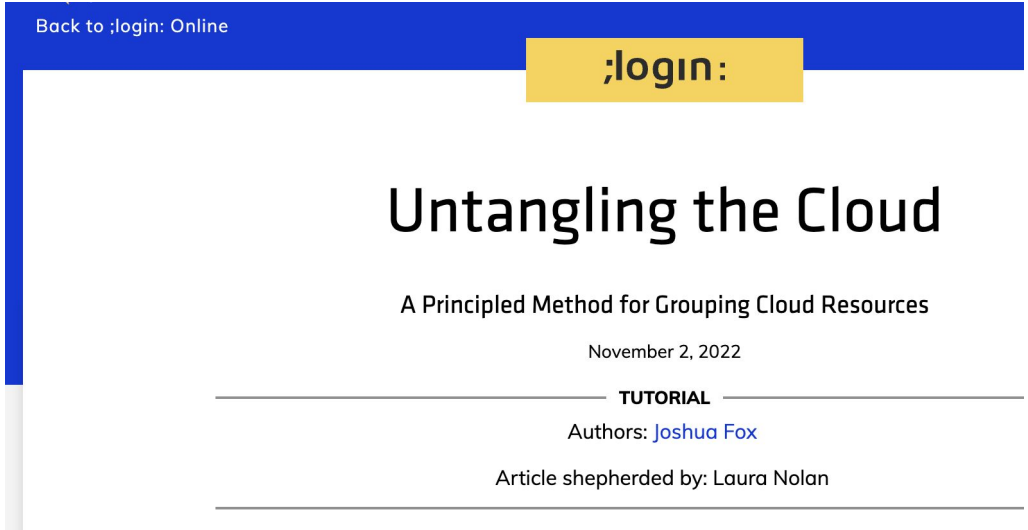
What you will see

A still-useful **25-year old** software tool for Java development,

reimagined for the cloud.

:Login; article

bit.ly/joshua-usenix


Back to ;login: Online

;login:

Untangling the Cloud

A Principled Method for Grouping Cloud Resources

November 2, 2022

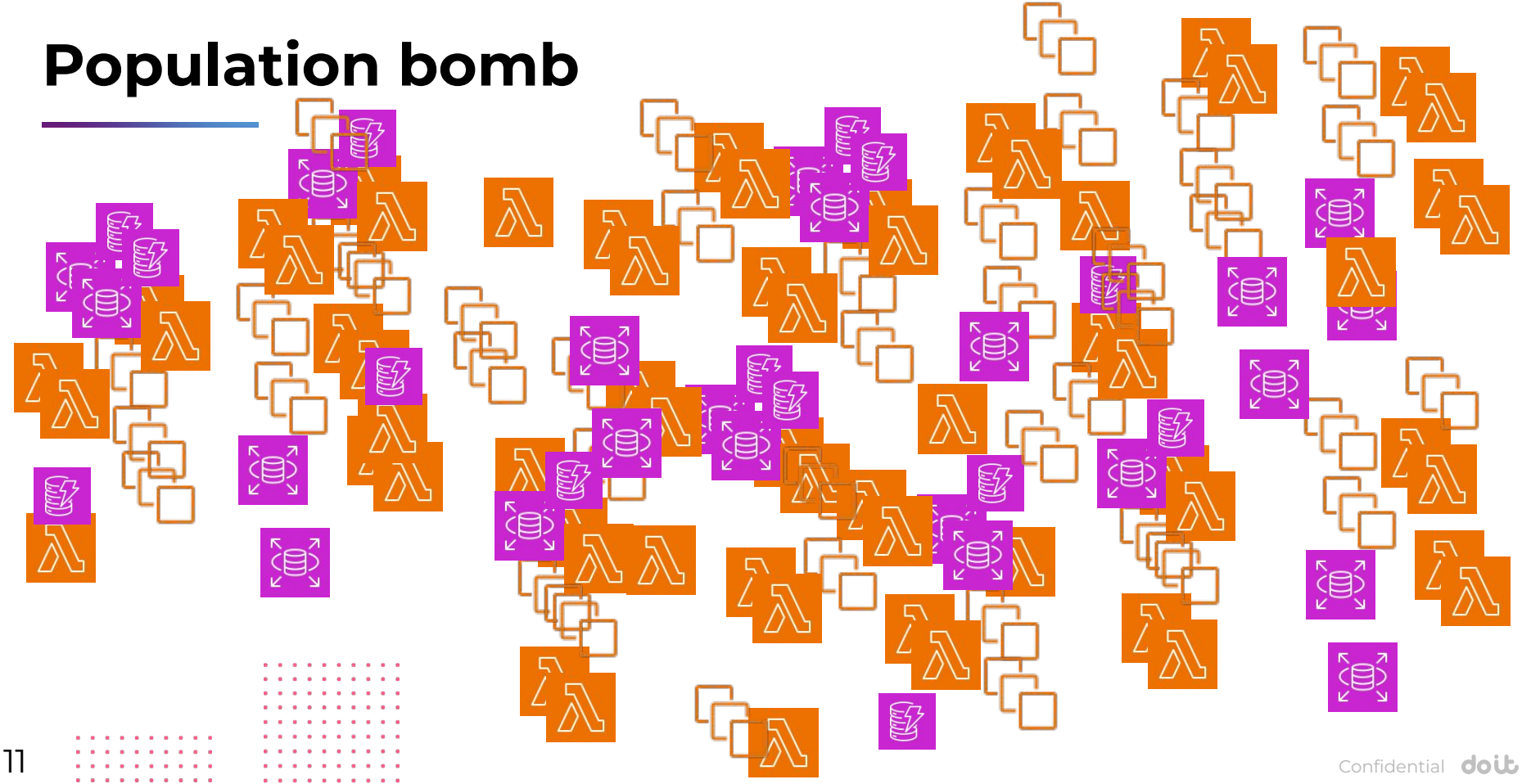
TUTORIAL

Authors: Joshua Fox

Article shepherded by: Laura Nolan

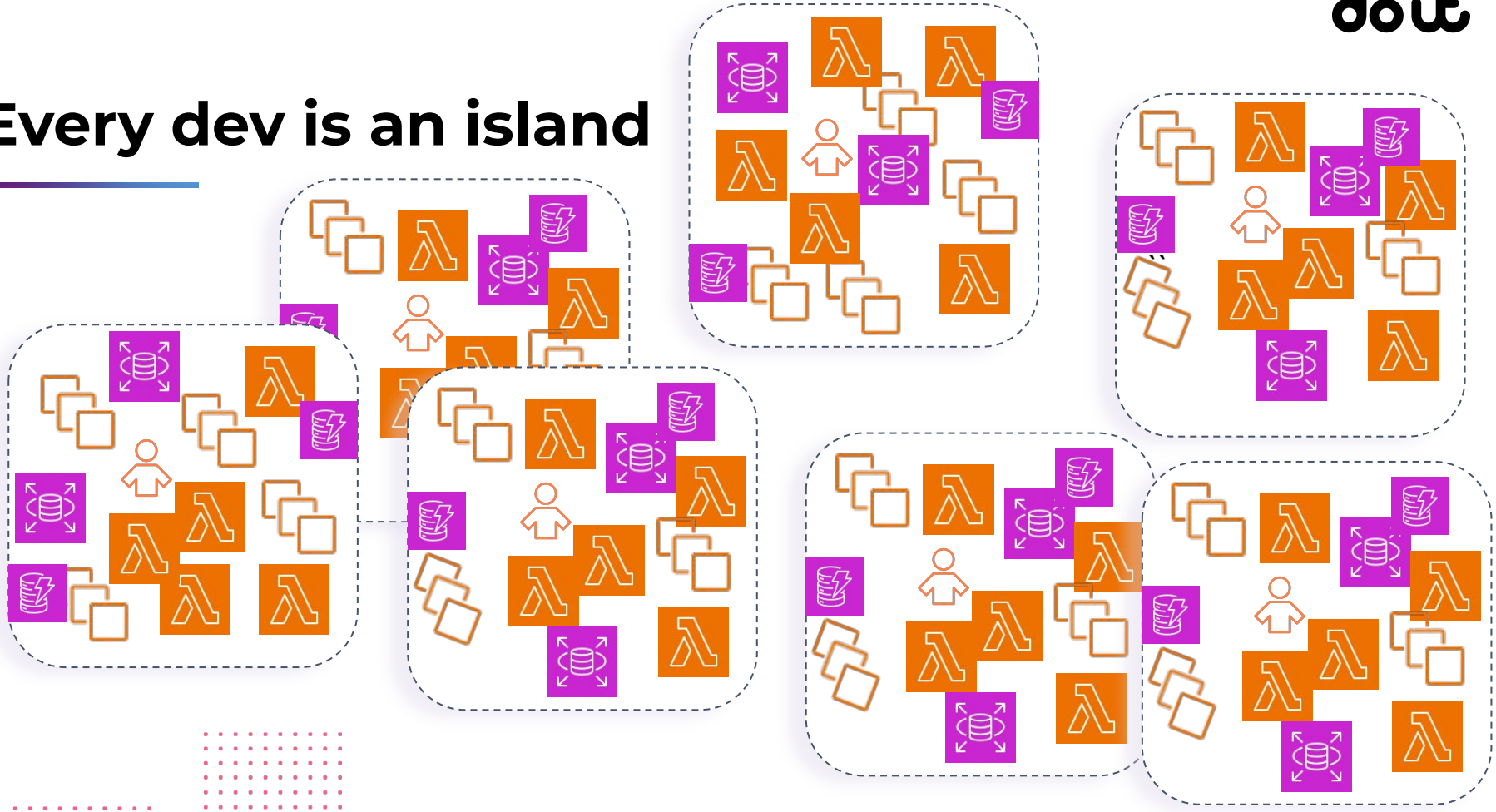
What my customers struggle with **Case Studies**

Population bomb

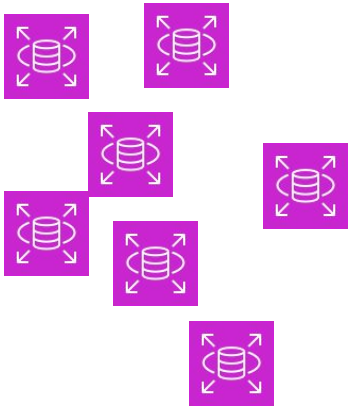


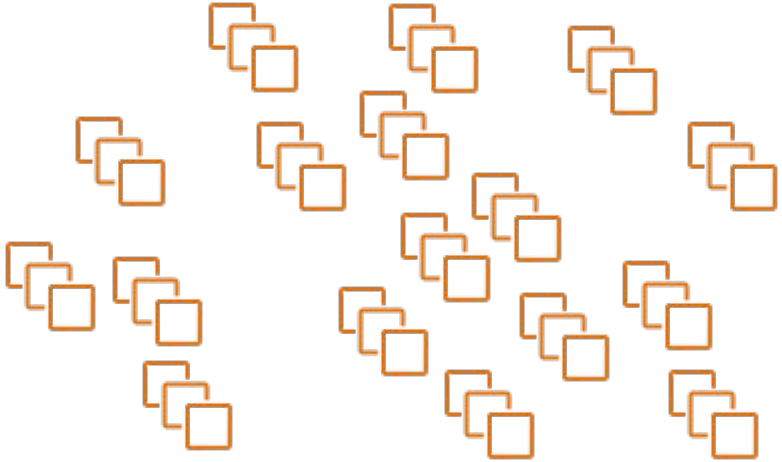
What I do

Every dev is an island



The estates

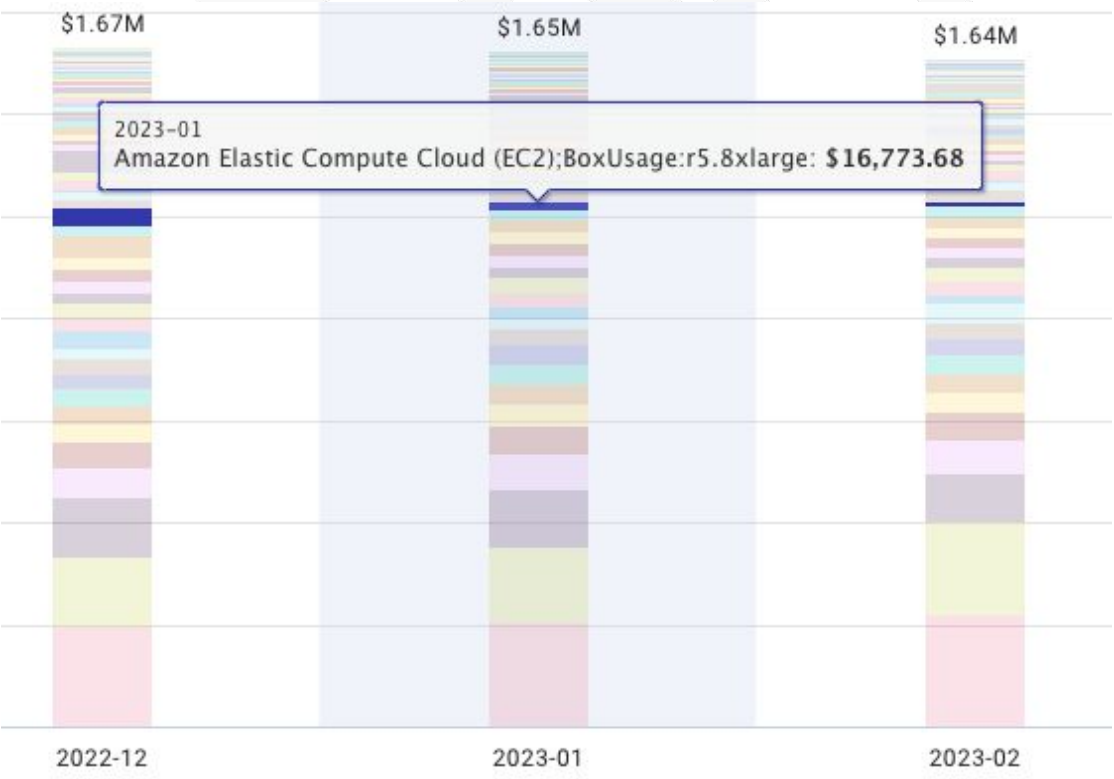




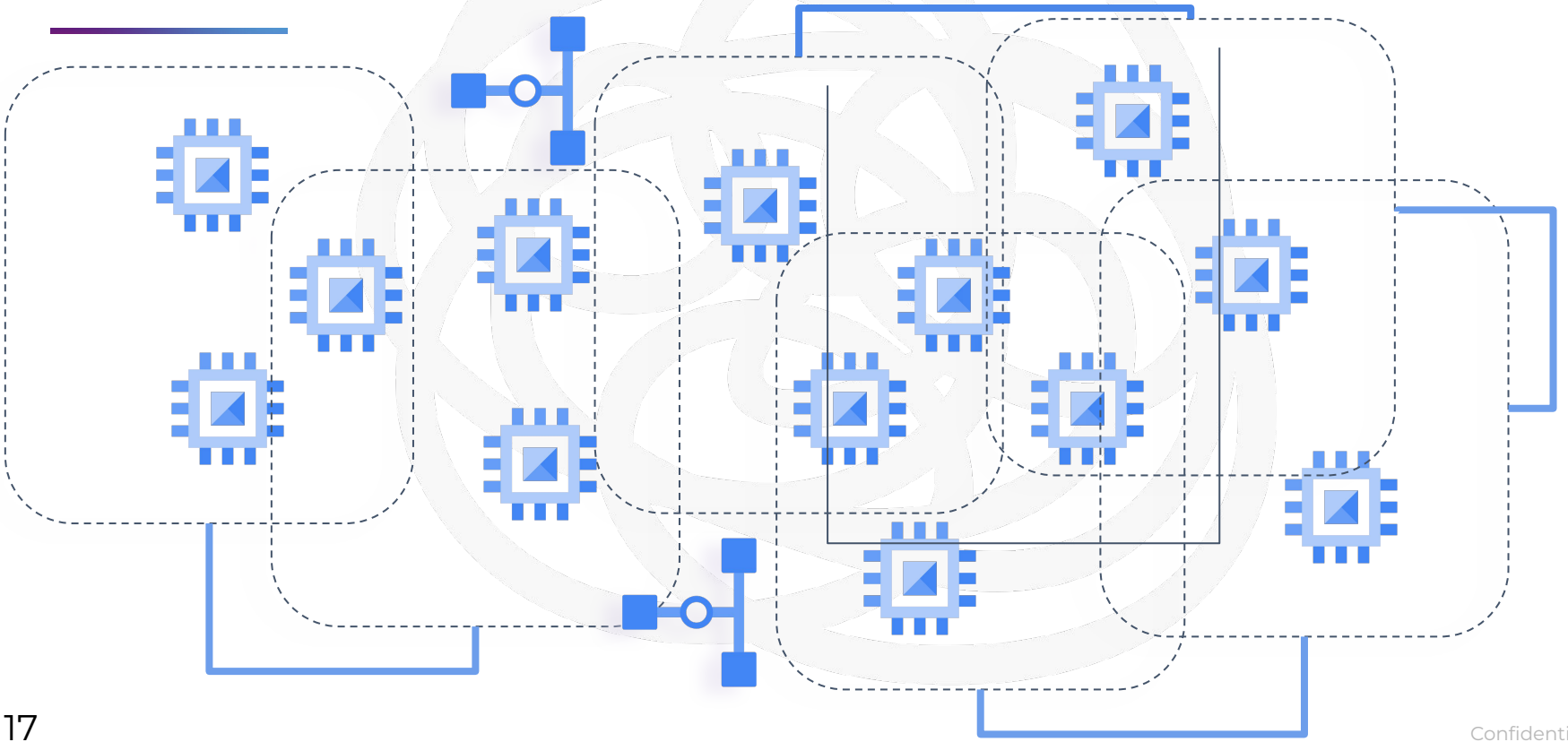
How it hurts

Let me count the ways

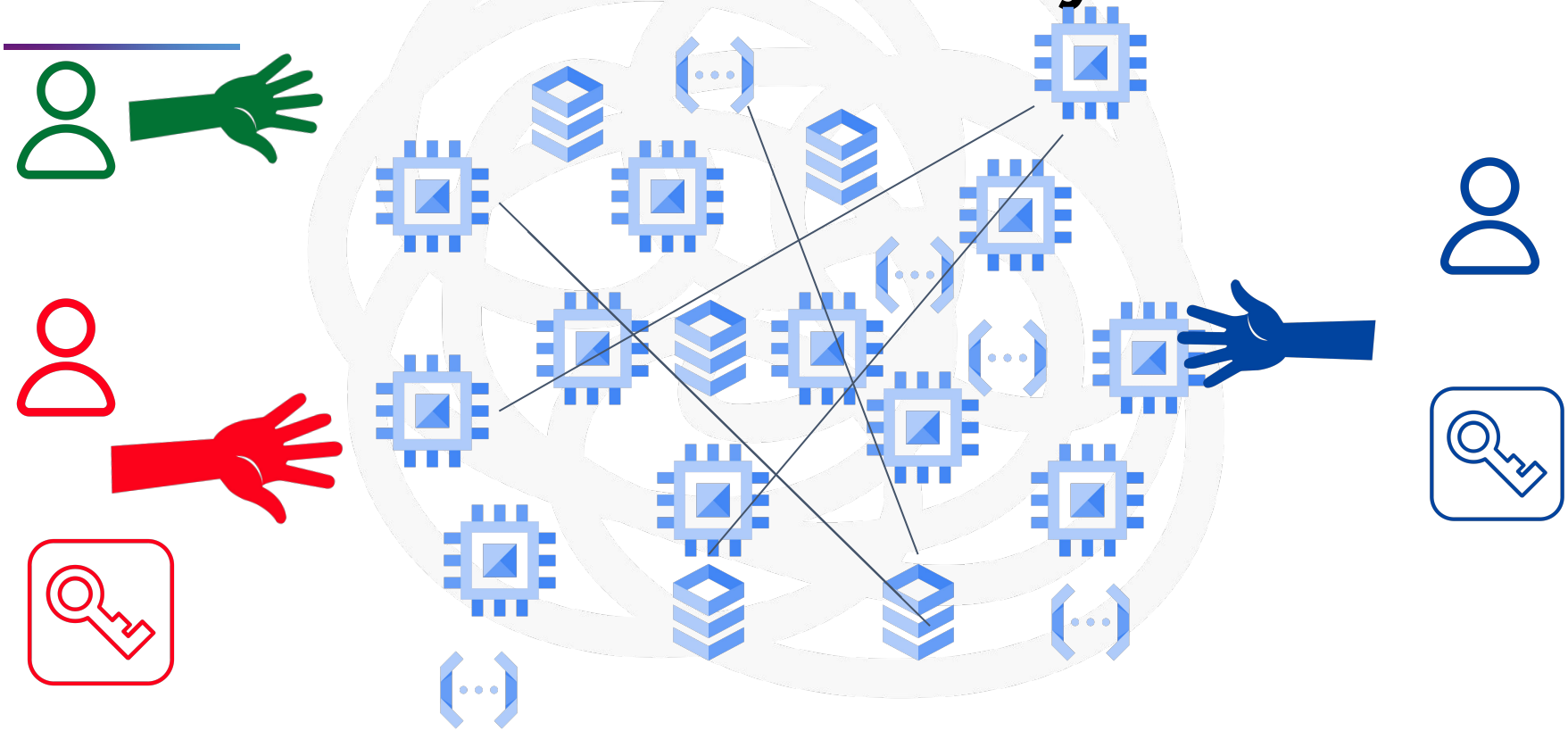
Just can’t deal with cost



Just can’t deal with network security



Just can’t deal with IAM security



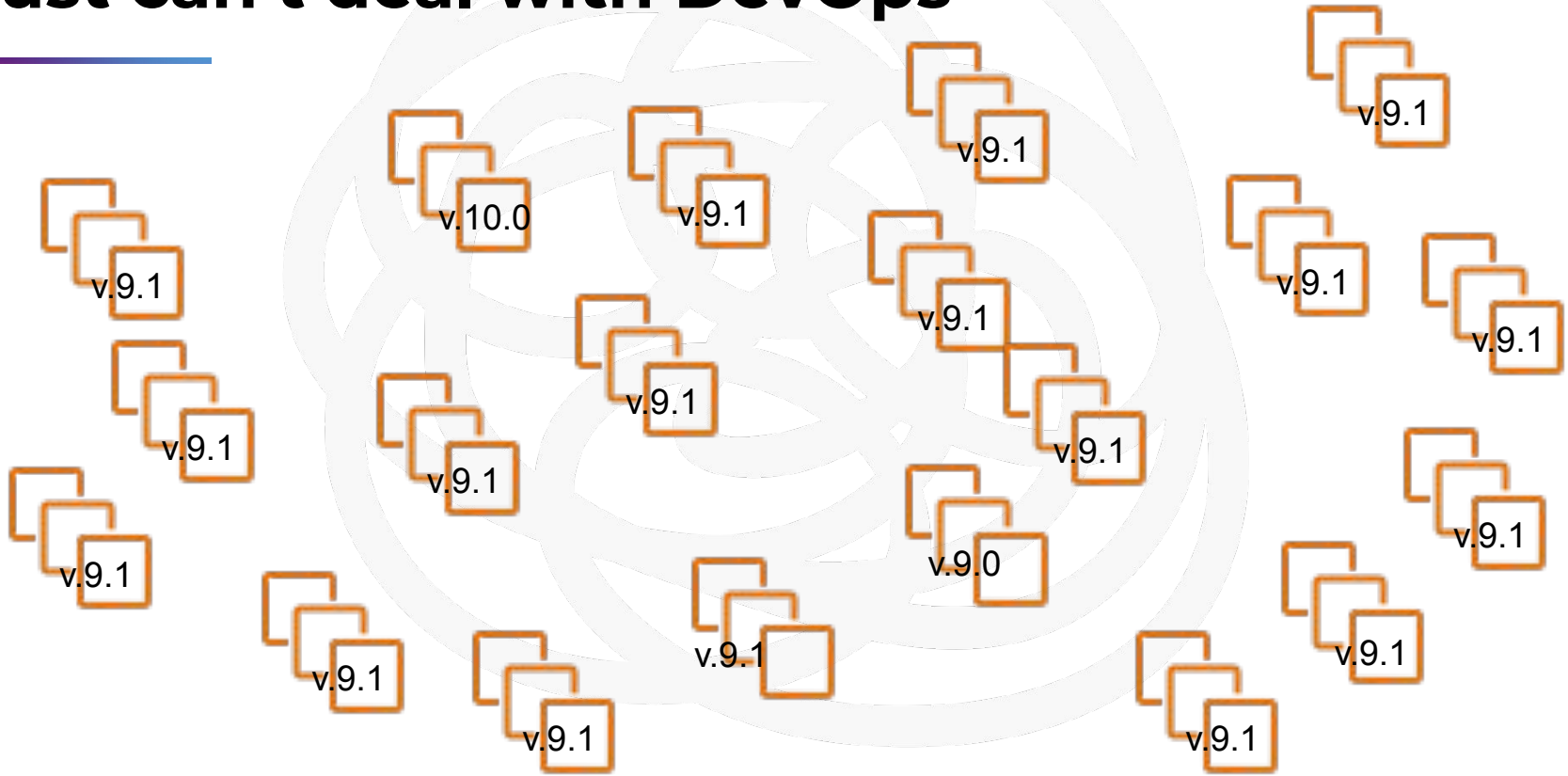
Just can’t deal with change

```
{...
    "type": "object",
    "properties": {...
        "age": {
            "type": "number",
            "minimum": 13
        }
    },
    "required": [   ...
        "age"
    ]
}
```

```
{...
    "type": "object",
    "properties": {...
        "birthdate": {
            "type": "string",
            "format": "date"
        }
    },
    "required": [   ...
        "birthdate"
    ]
}
```

X 100

Just can’t deal with DevOps



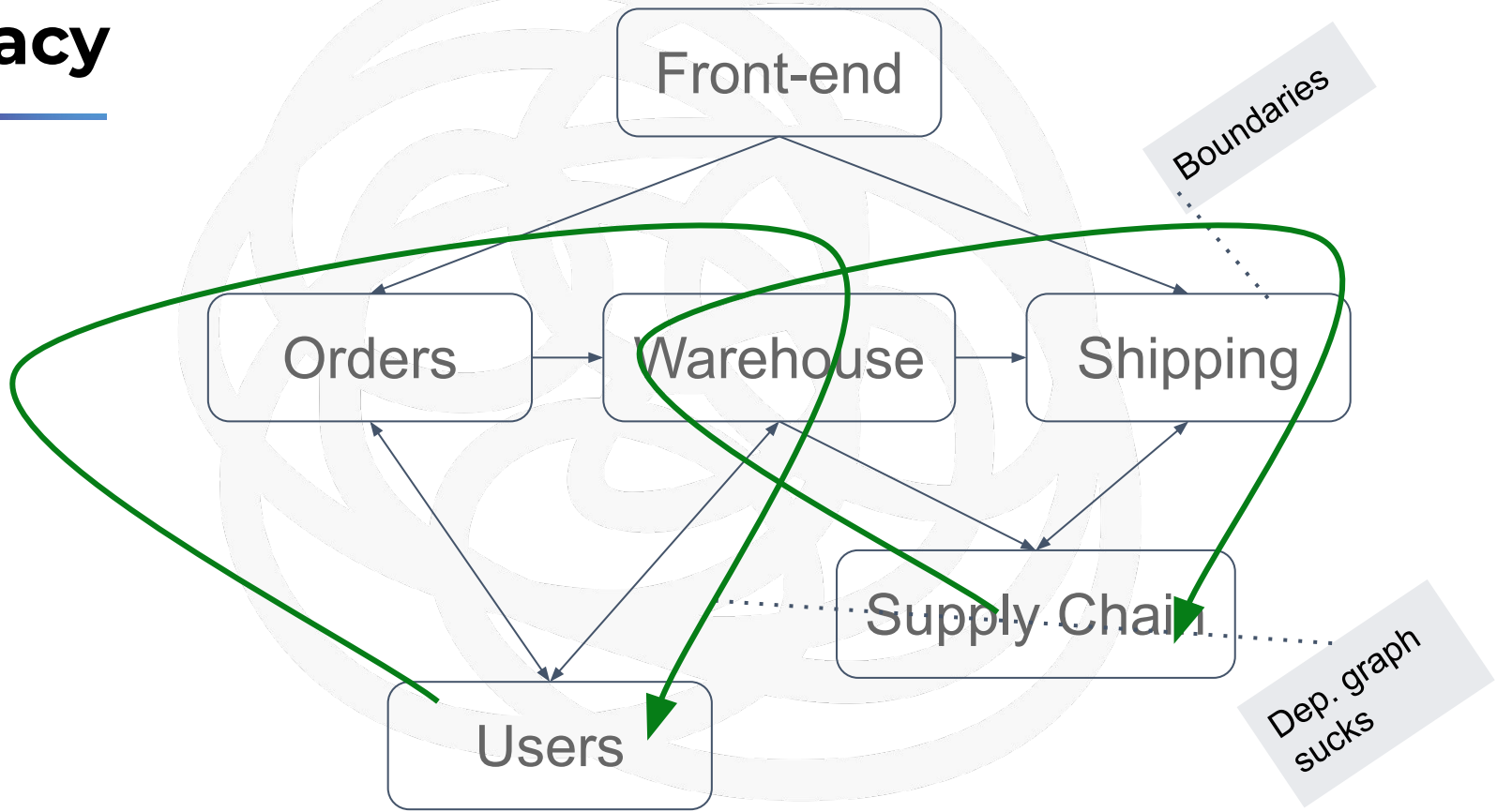
What a solution looks like

- Choose your boundaries
- … in a way that enables a clear, untangled graph of dependencies

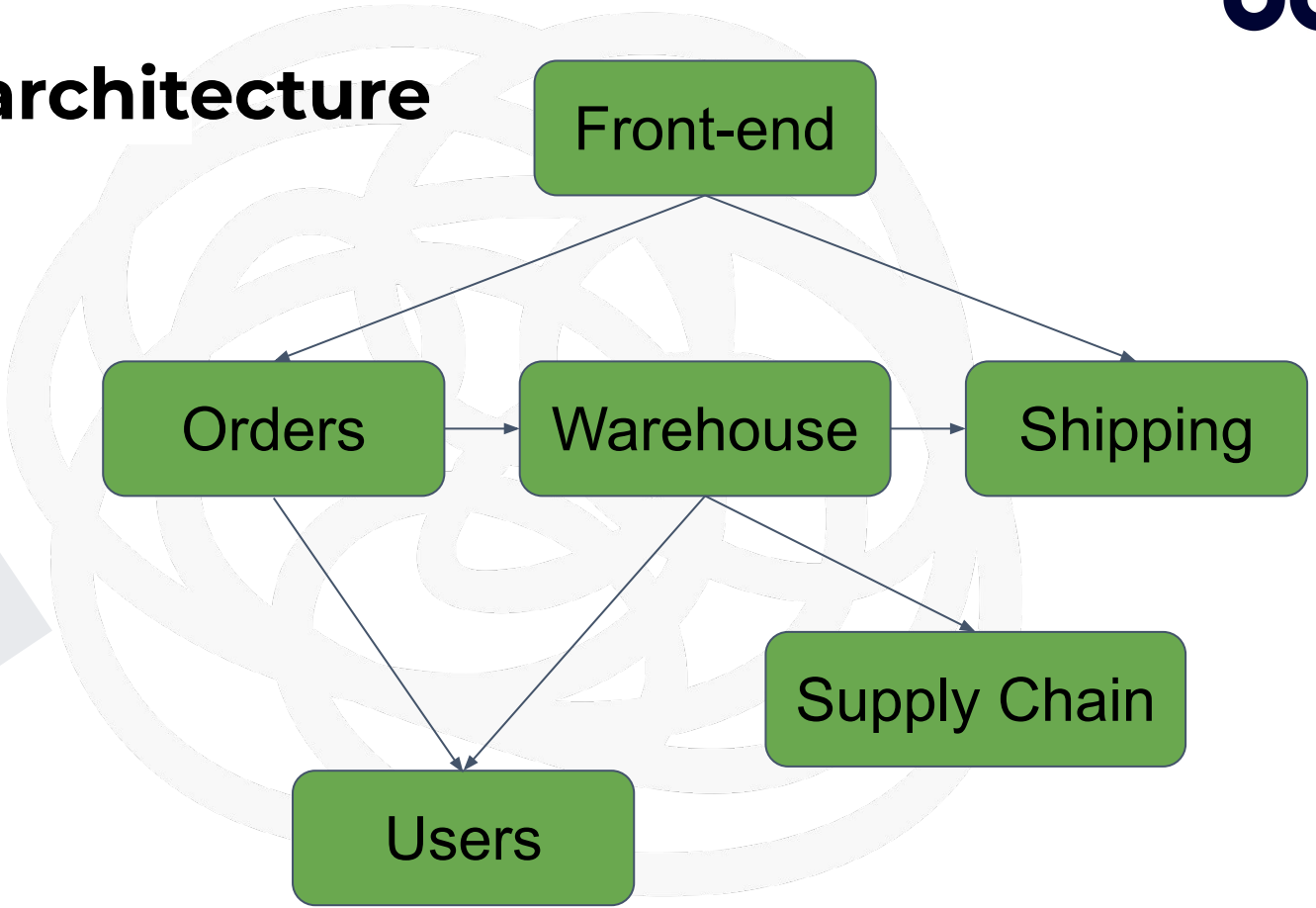
Case Study

E-commerce company gets its act together

Legacy

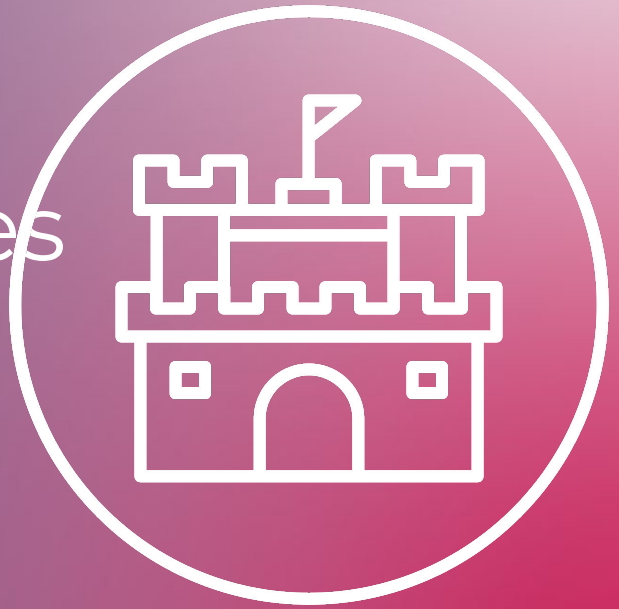


After Rearchitecture



Boundaries define a beautiful dep graph

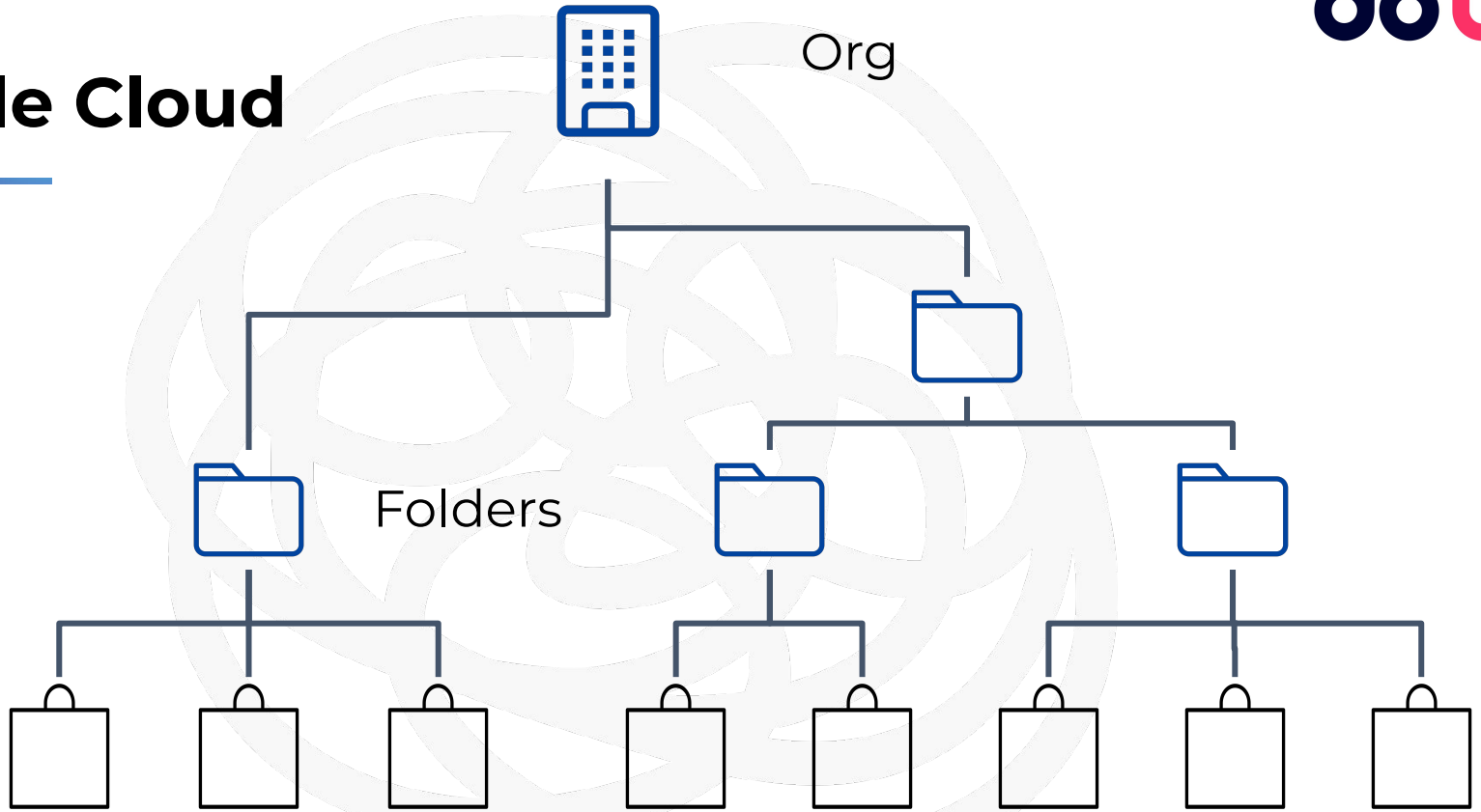
What kind of Boundaries
Different tools to use



Units

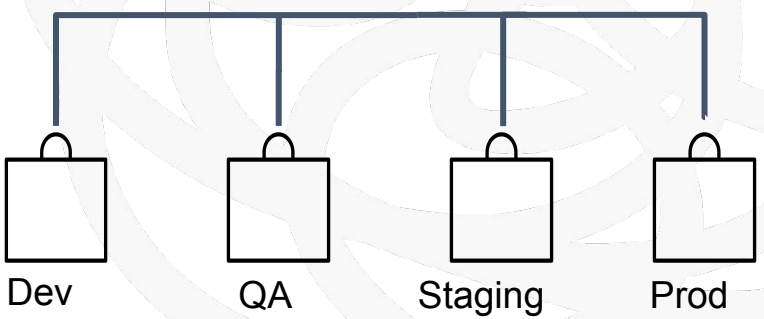
- Cloud Resources bundled into on a higher level: Microservices and apps
- Tools for that…

Google Cloud



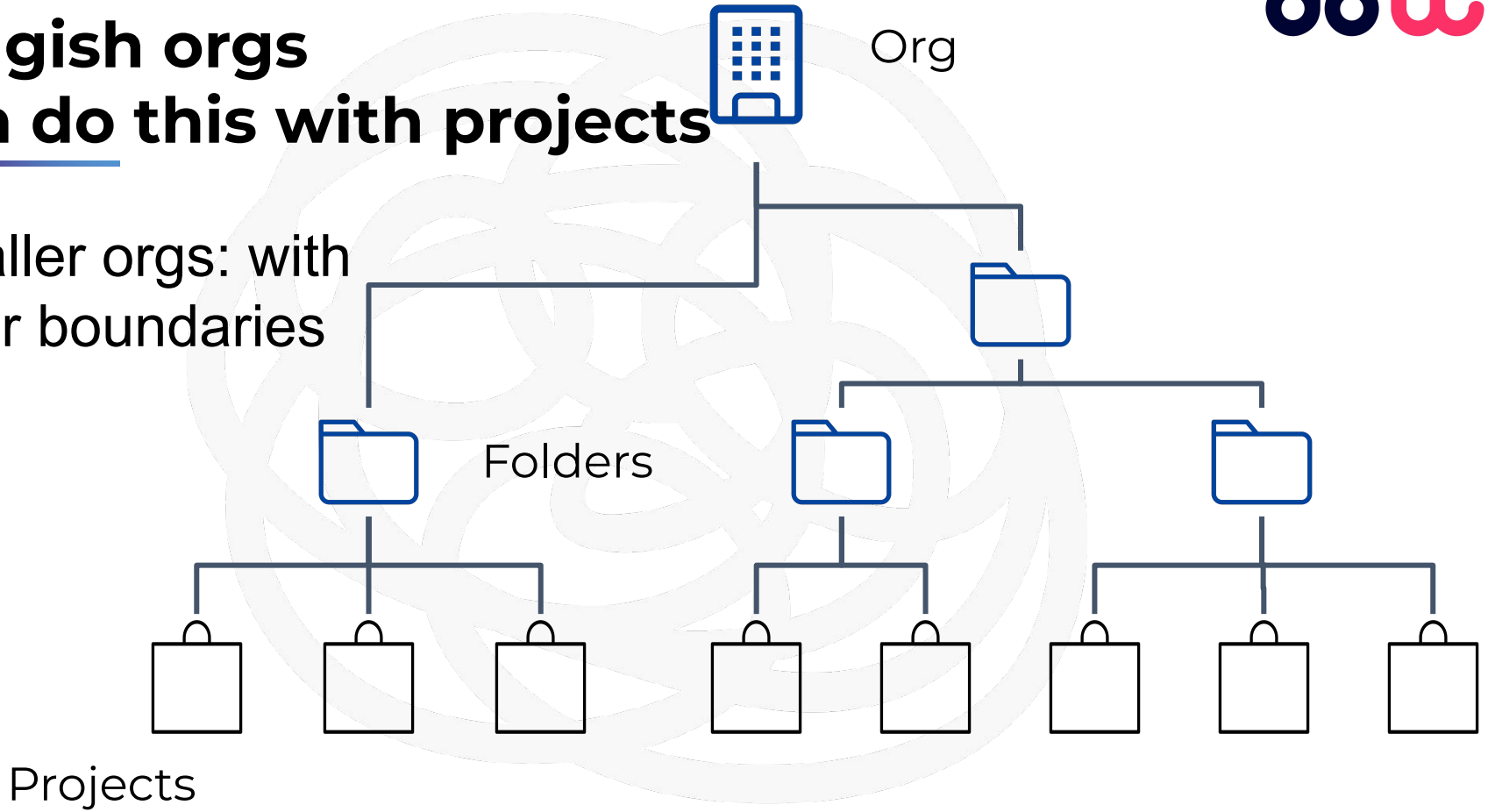
Projects

Out of today's scope

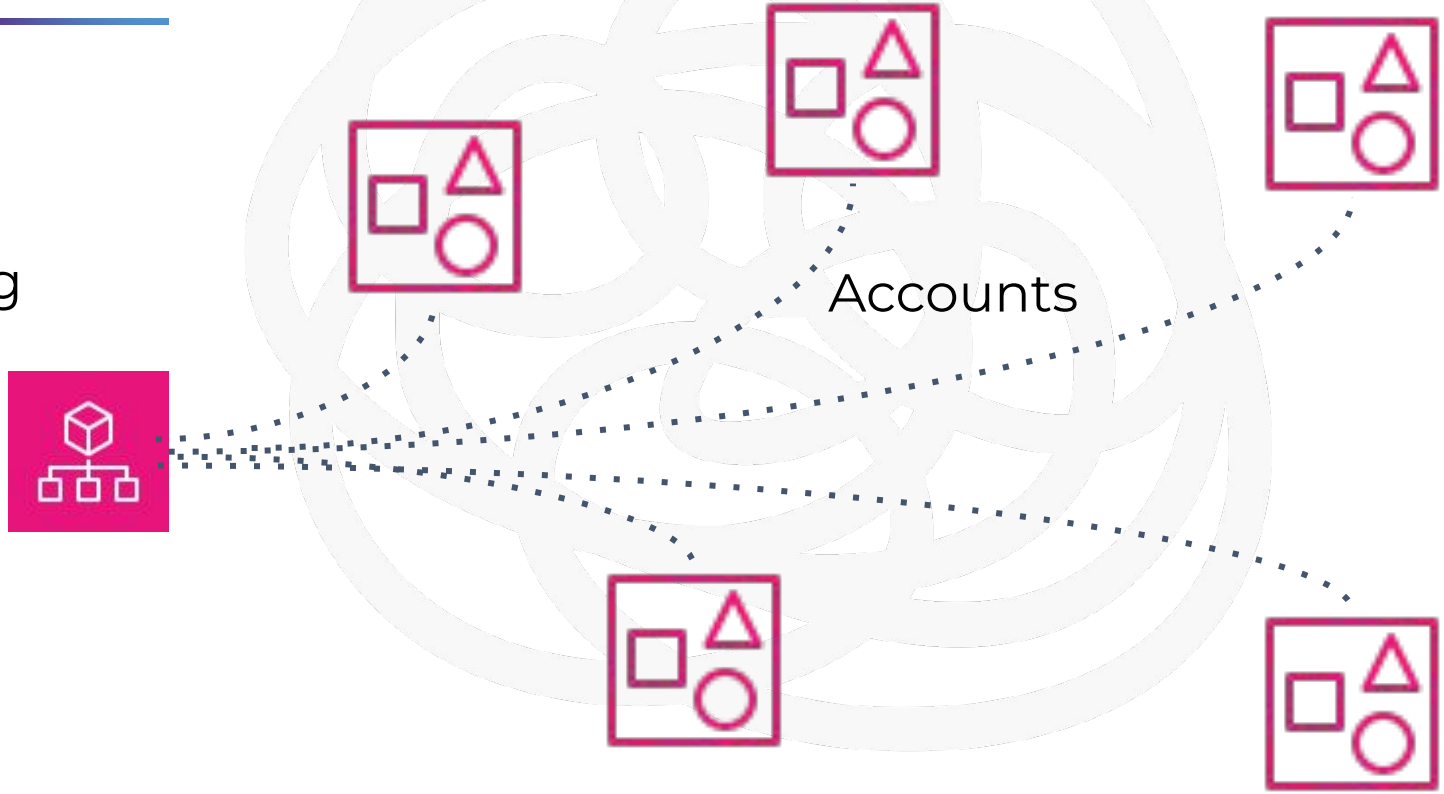


Biggish orgs can do this with projects

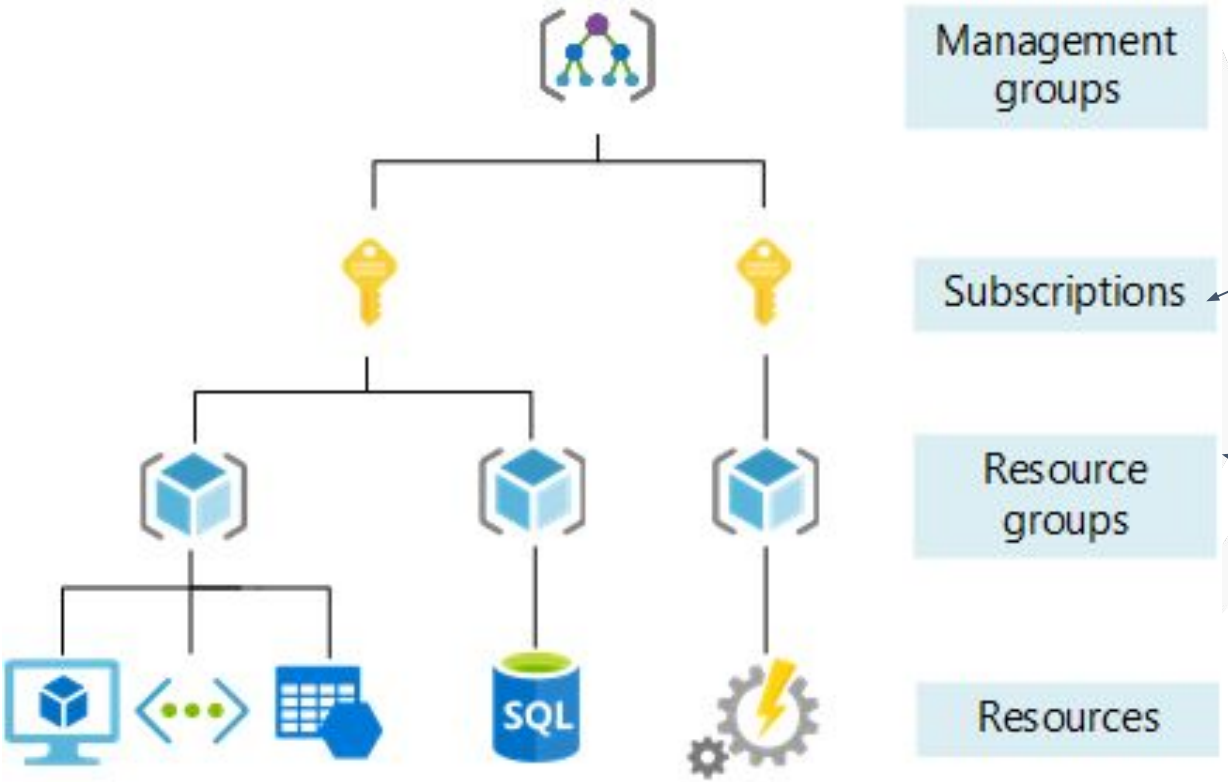
Smaller orgs: with other boundaries



AWS



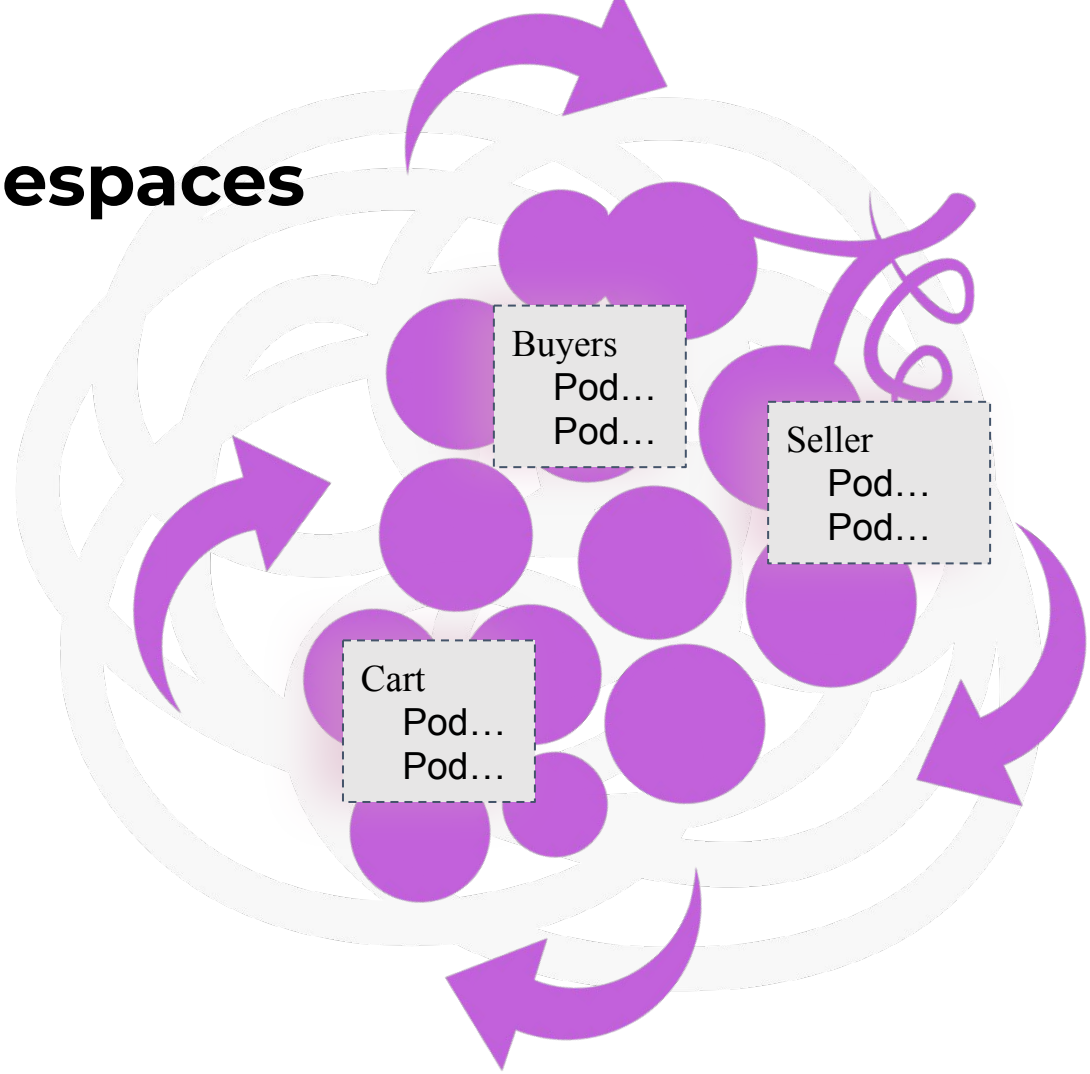
Azure



~Project/Accounts, but generally fewer are used

Lightweight unit for boundaries

K8s namespaces



Large cluster or lots of small ones?

The problem with labels

T.M.I.

Labels

region: eu-central-1
tier: back-end
business-unit: video

region: us-west-2
tier: backend
business-unit: ecommerce

region: us-west-2
tier: back-end
business-unit:
ecommerce

region: us-east-1
tier: front-end
business-unit: video

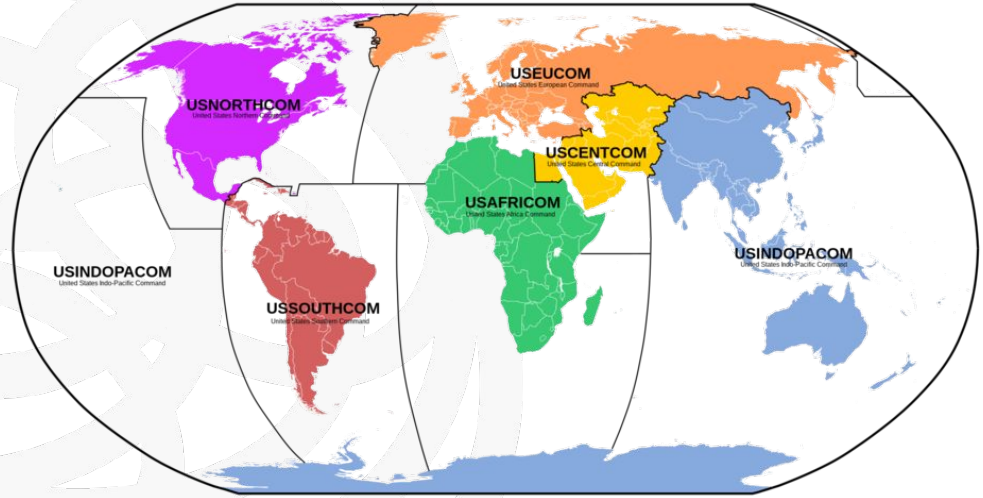
region: us-west-1
tier: backend
business-unit: ecommerce

region: eu-central-1
tier: batch
business-unit: video

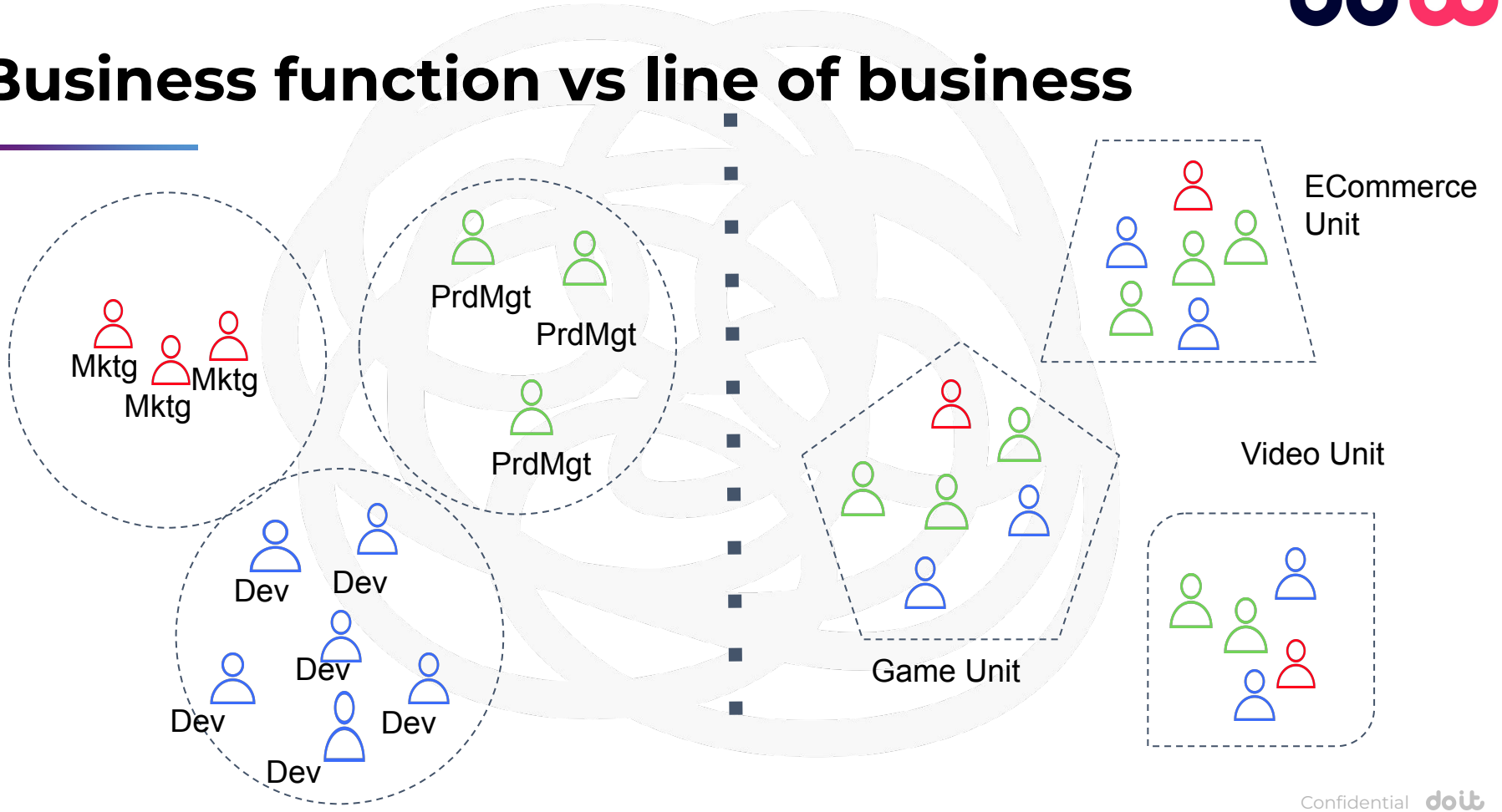
region: eu-central-1
tier: front-end
business-unit: gaming

region: us-east-1
tier: front-end
business-unit: video

Functional or Geo?

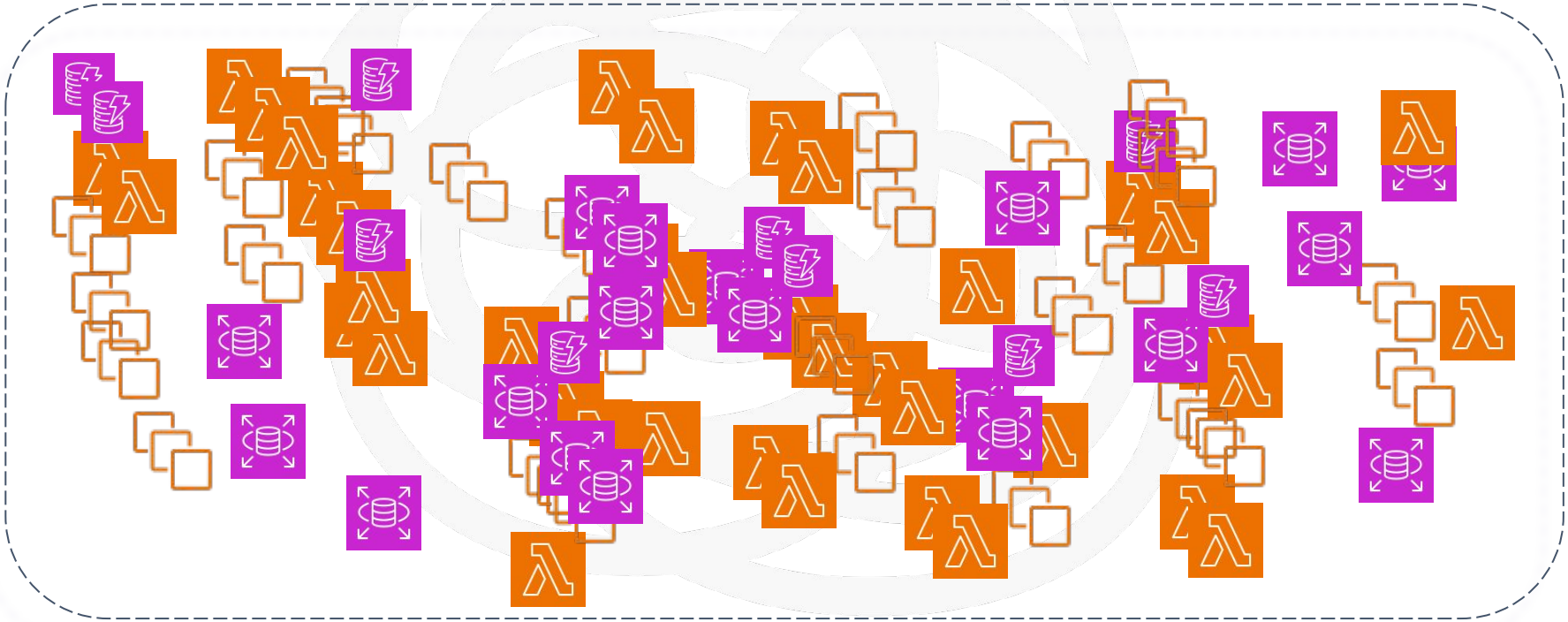


Business function vs line of business

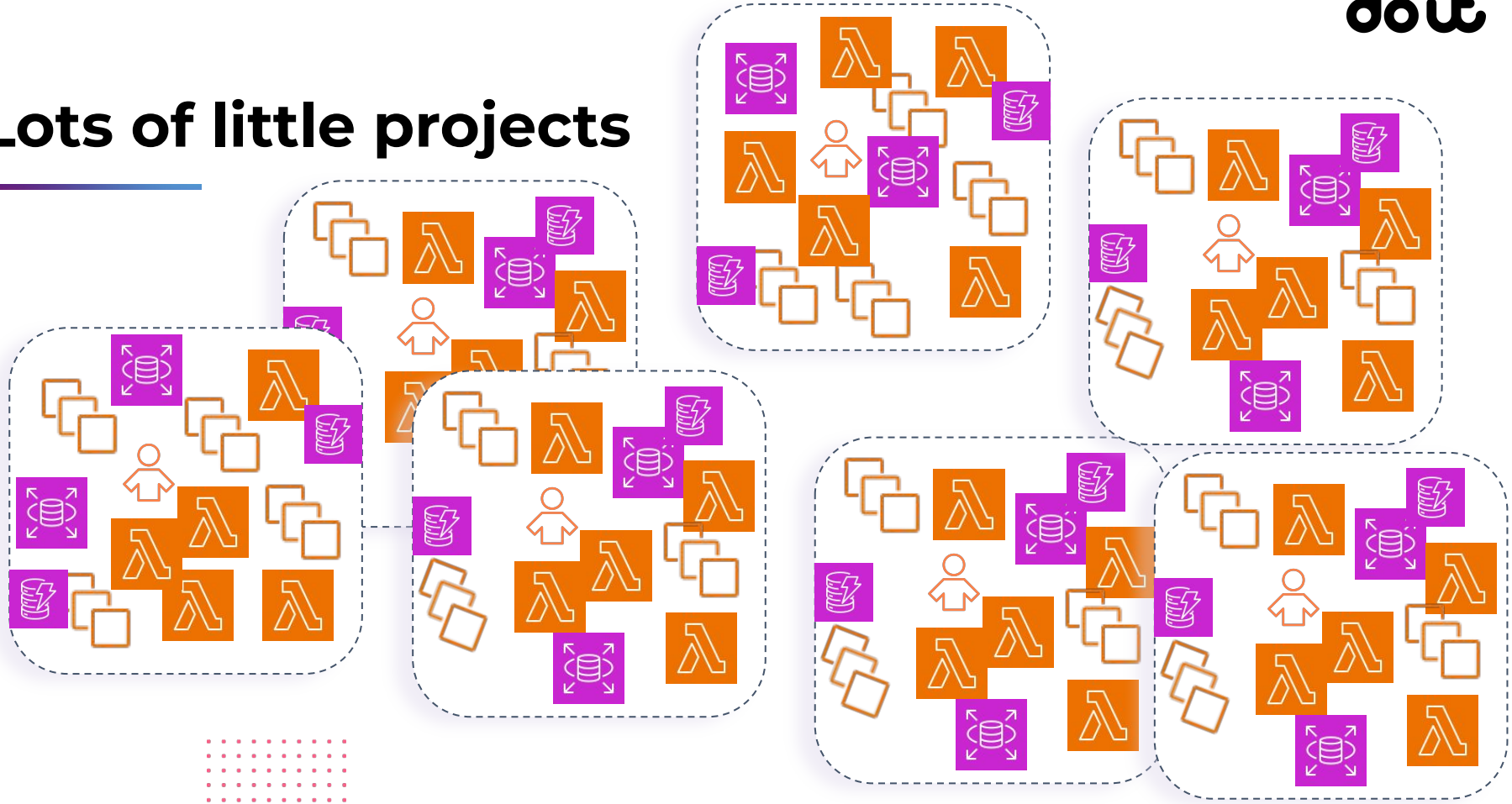


Principles for drawing boundaries
First: Failure modes

Megablob



Lots of little projects

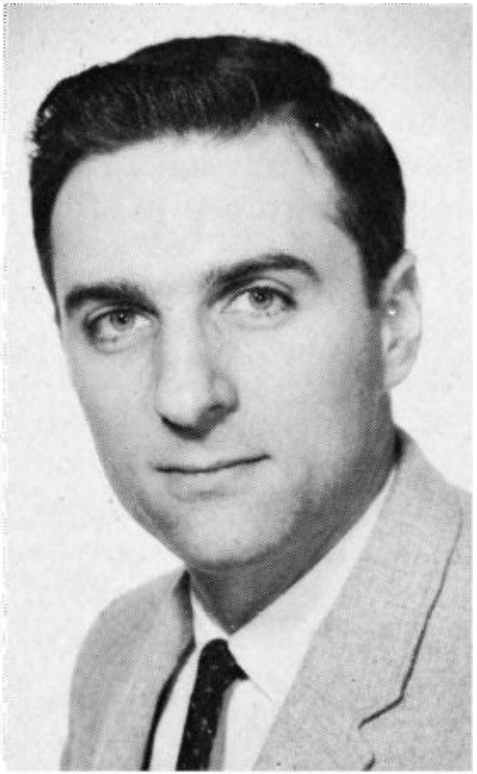


First guideline
Follow the business

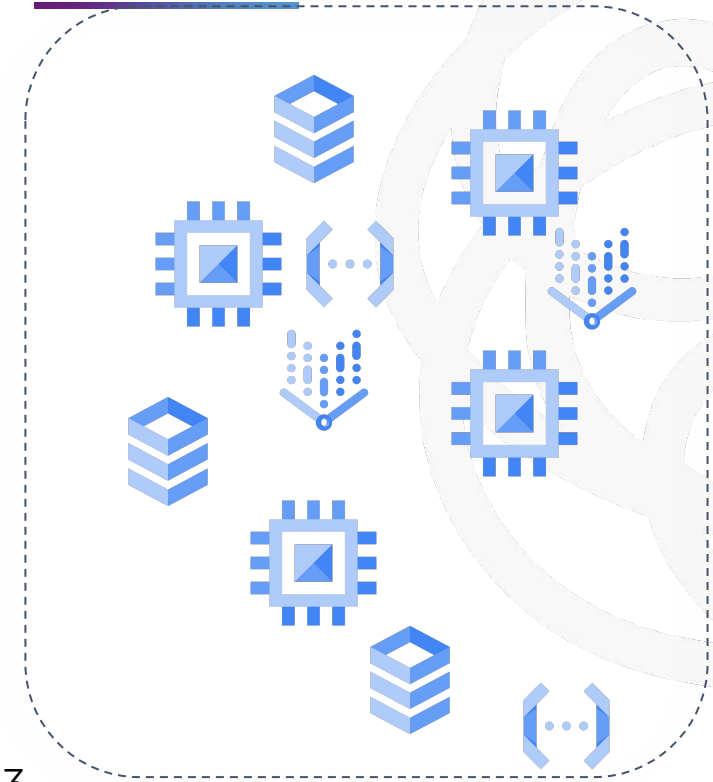
Conway's Law

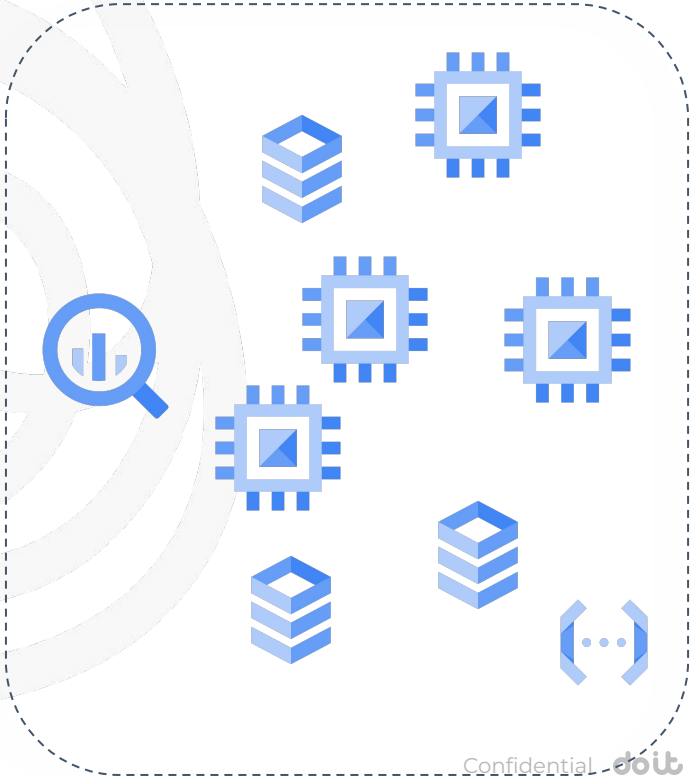
Melvin Conway:

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



Office in Ukraine and Boston





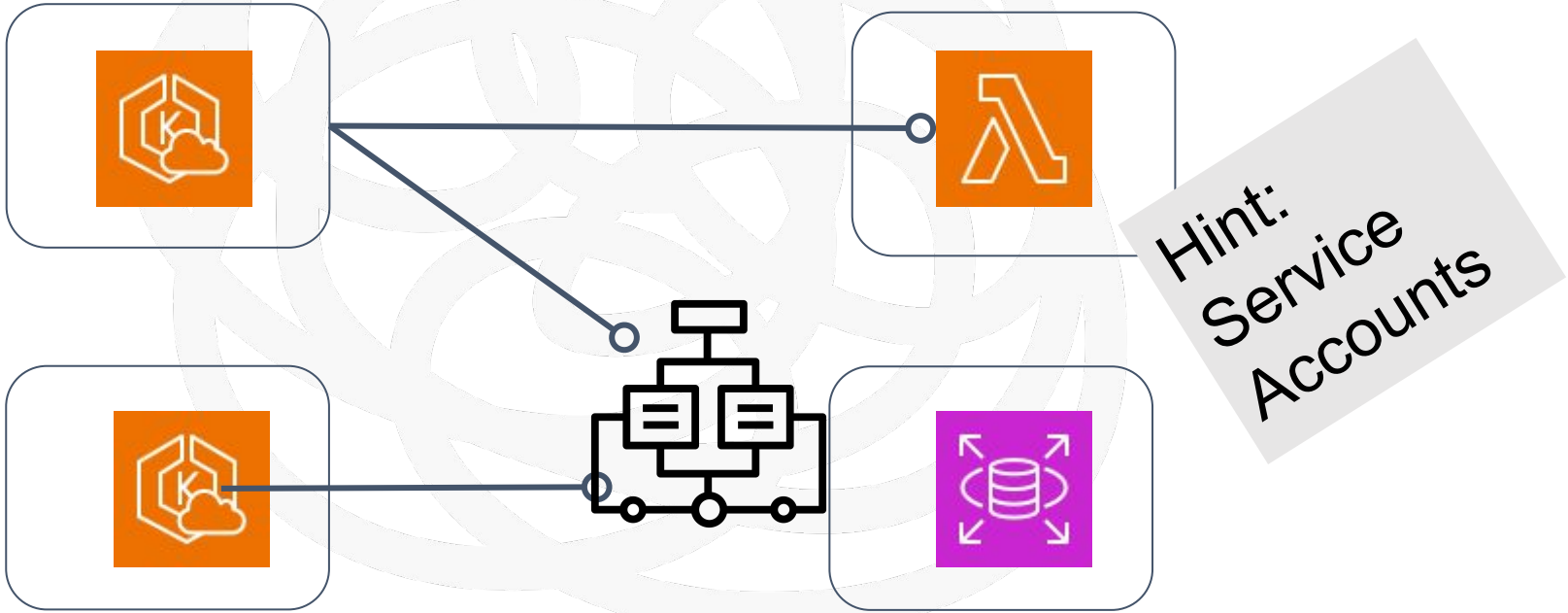
Iron Law: This *will* happen

- So make the architecture follow the organization
- And if you can, make the organization follow the necessary architecture.

The perfect graph
Rules for dependencies

What is a dependency

How is it defined technically?



General Concept: Code

requirements.txt

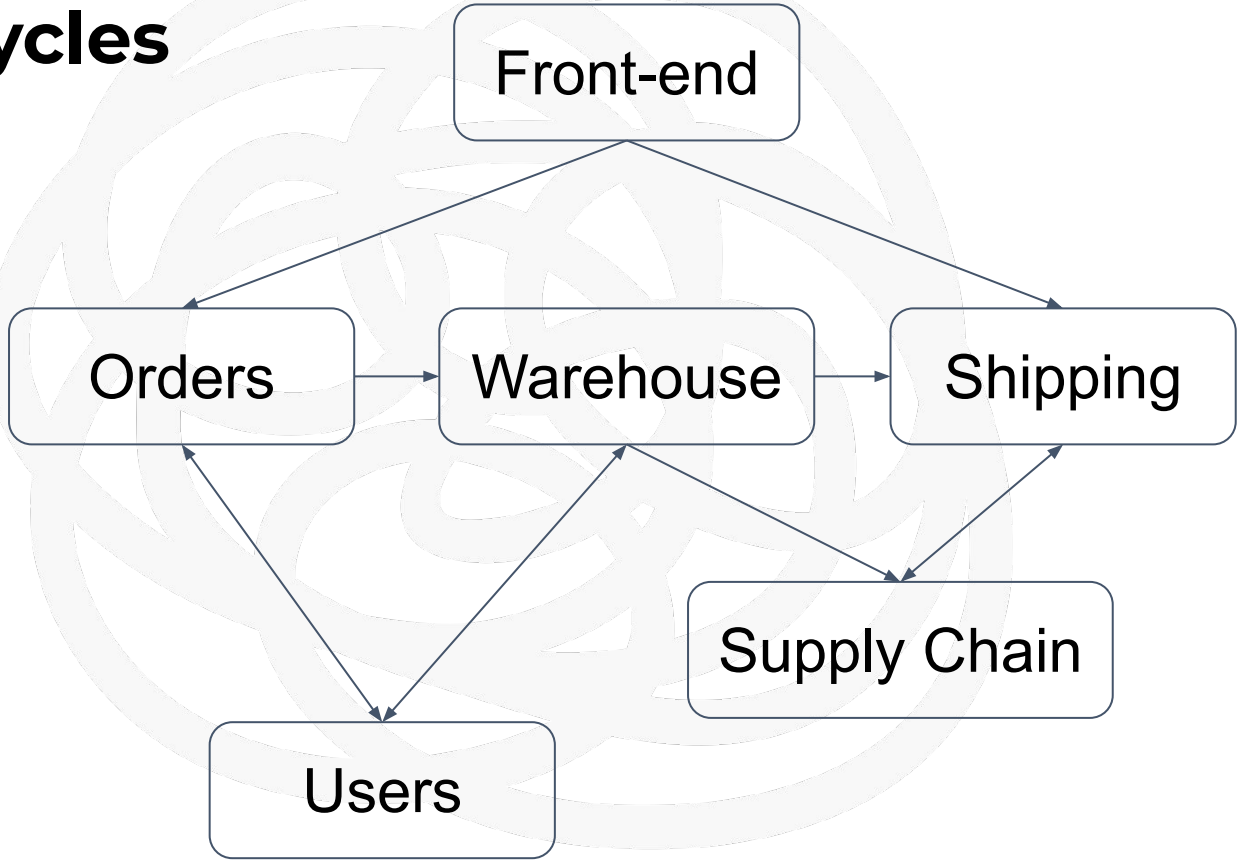
```
kleenexp
```

Shipping.java

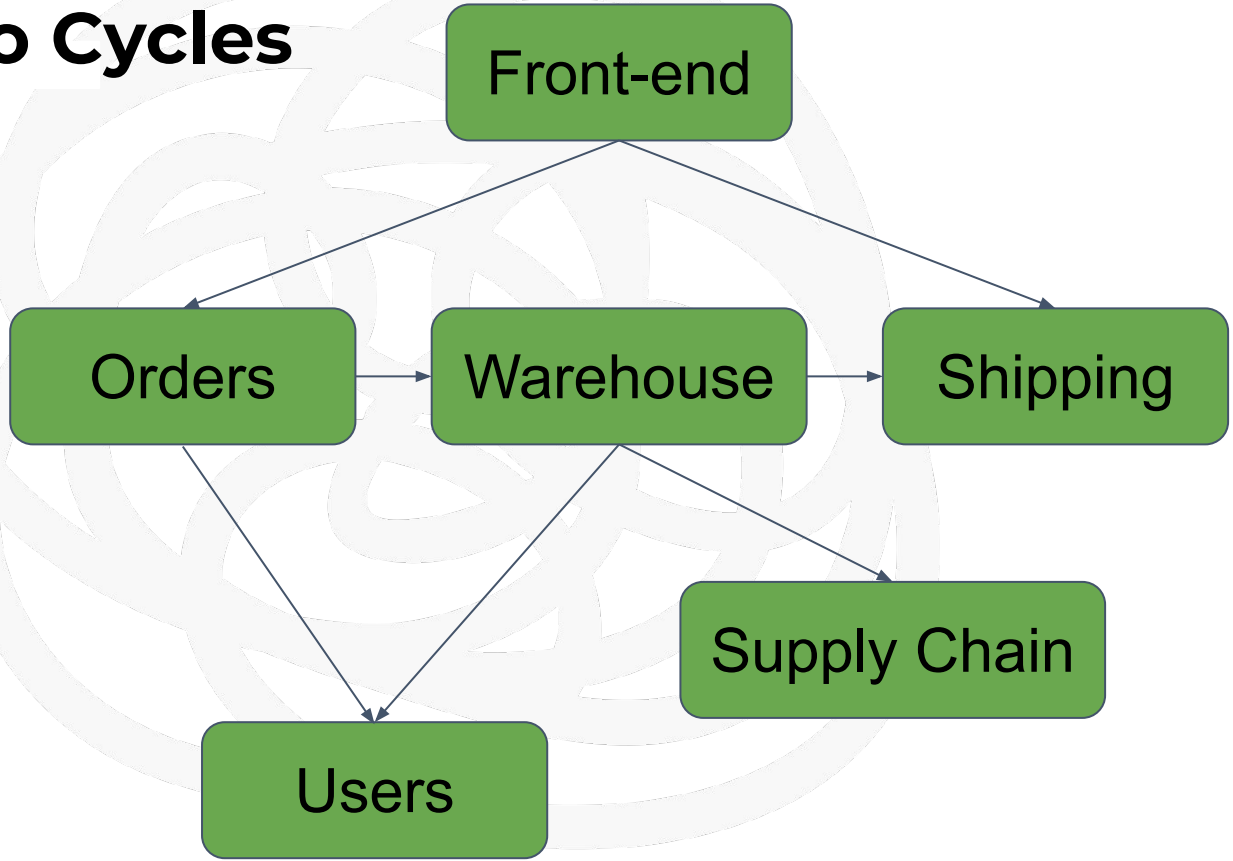
```
import org.apache.commons.io.IOUtils;

public class Shipping {
```

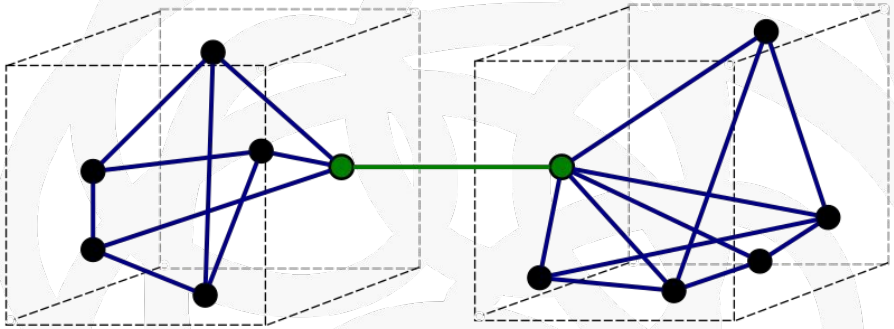
DAGit! Cycles



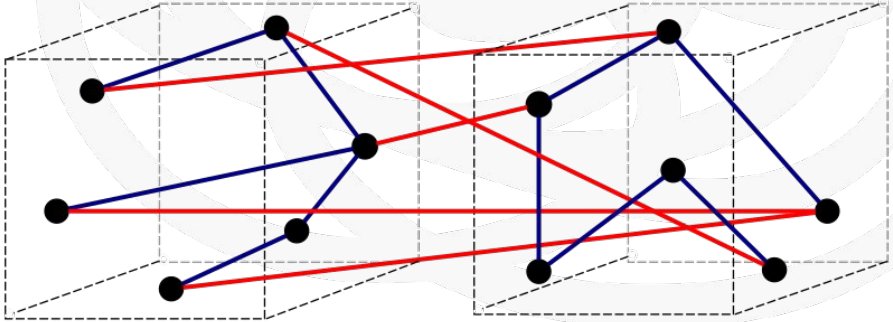
DAGged: No Cycles



Low Coupling, High Cohesion



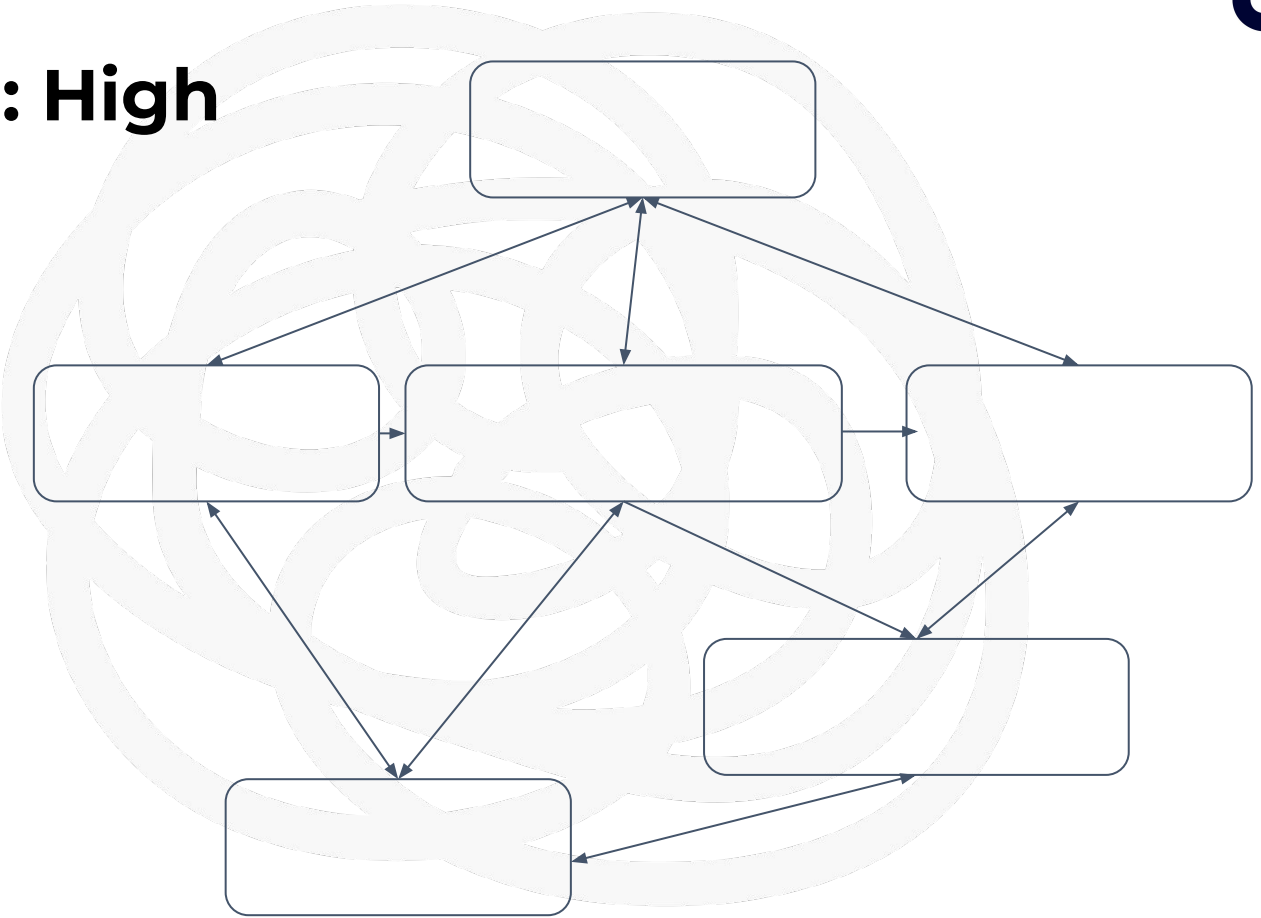
a) Good (loose coupling, high cohesion)



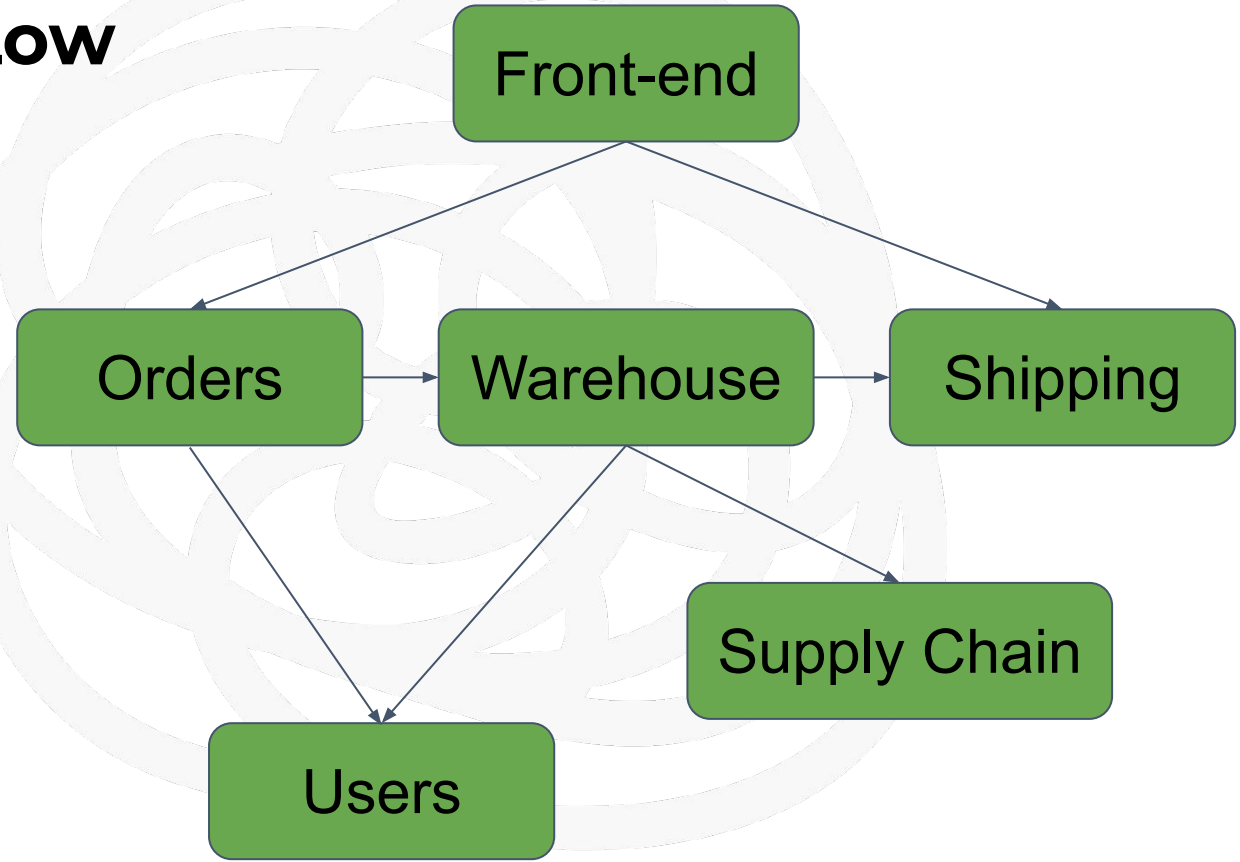
b) Bad (high coupling, low cohesion)

Wikipedia

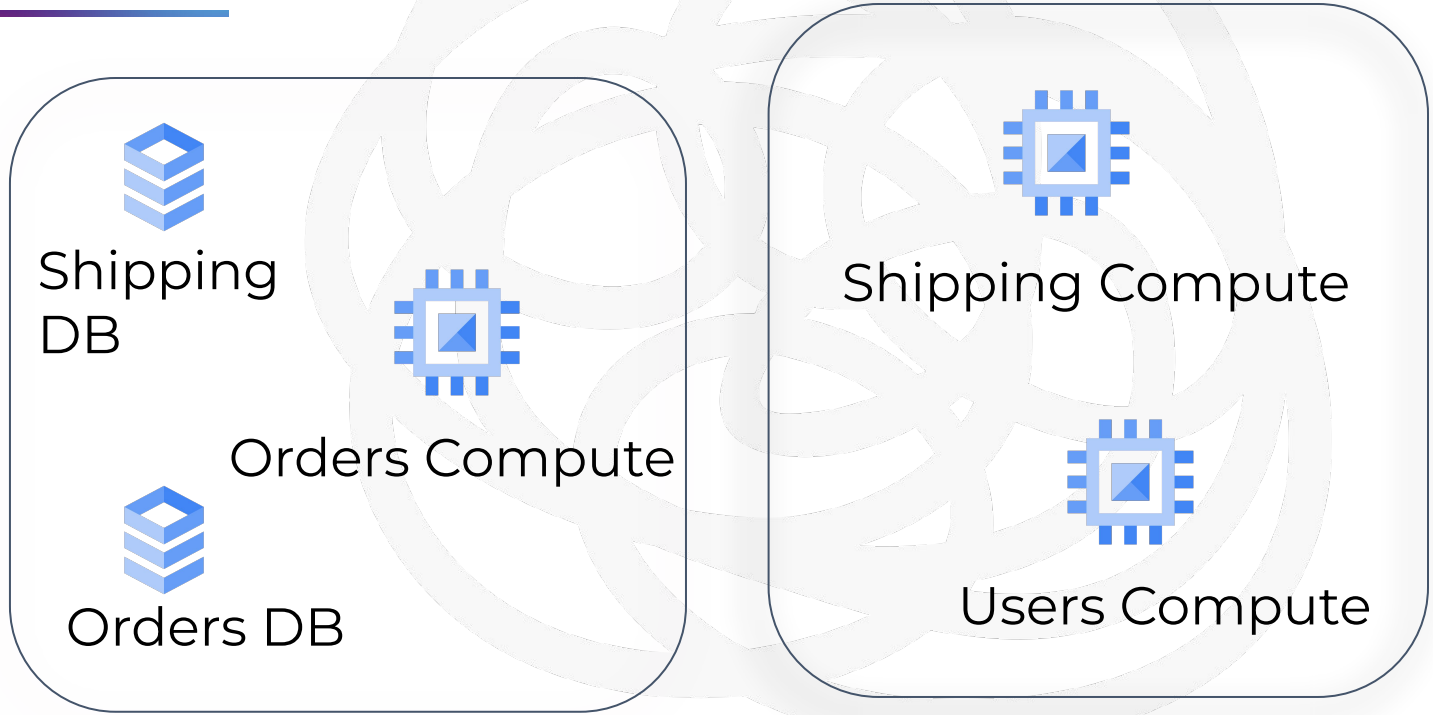
Coupling: High



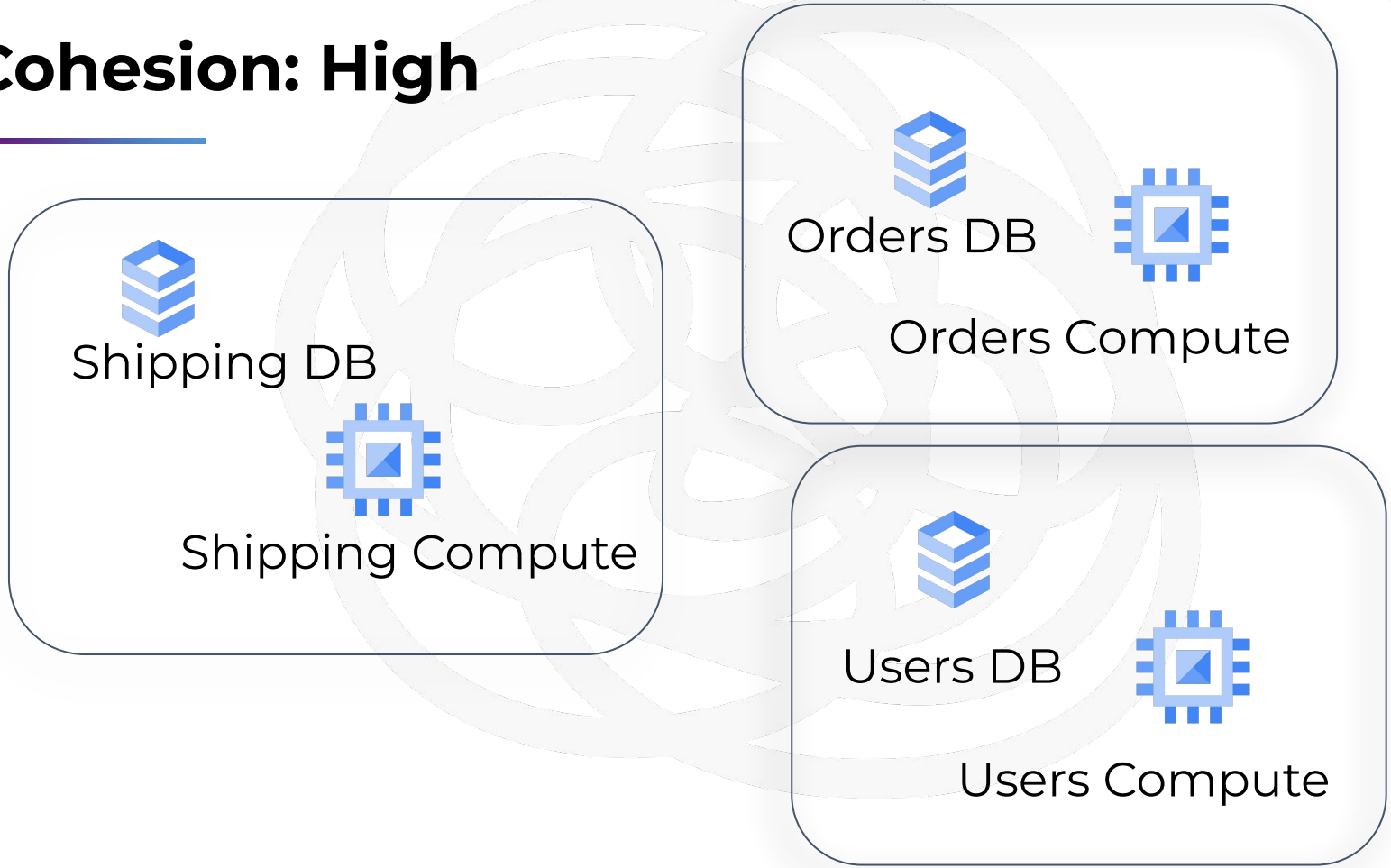
Coupling: Low



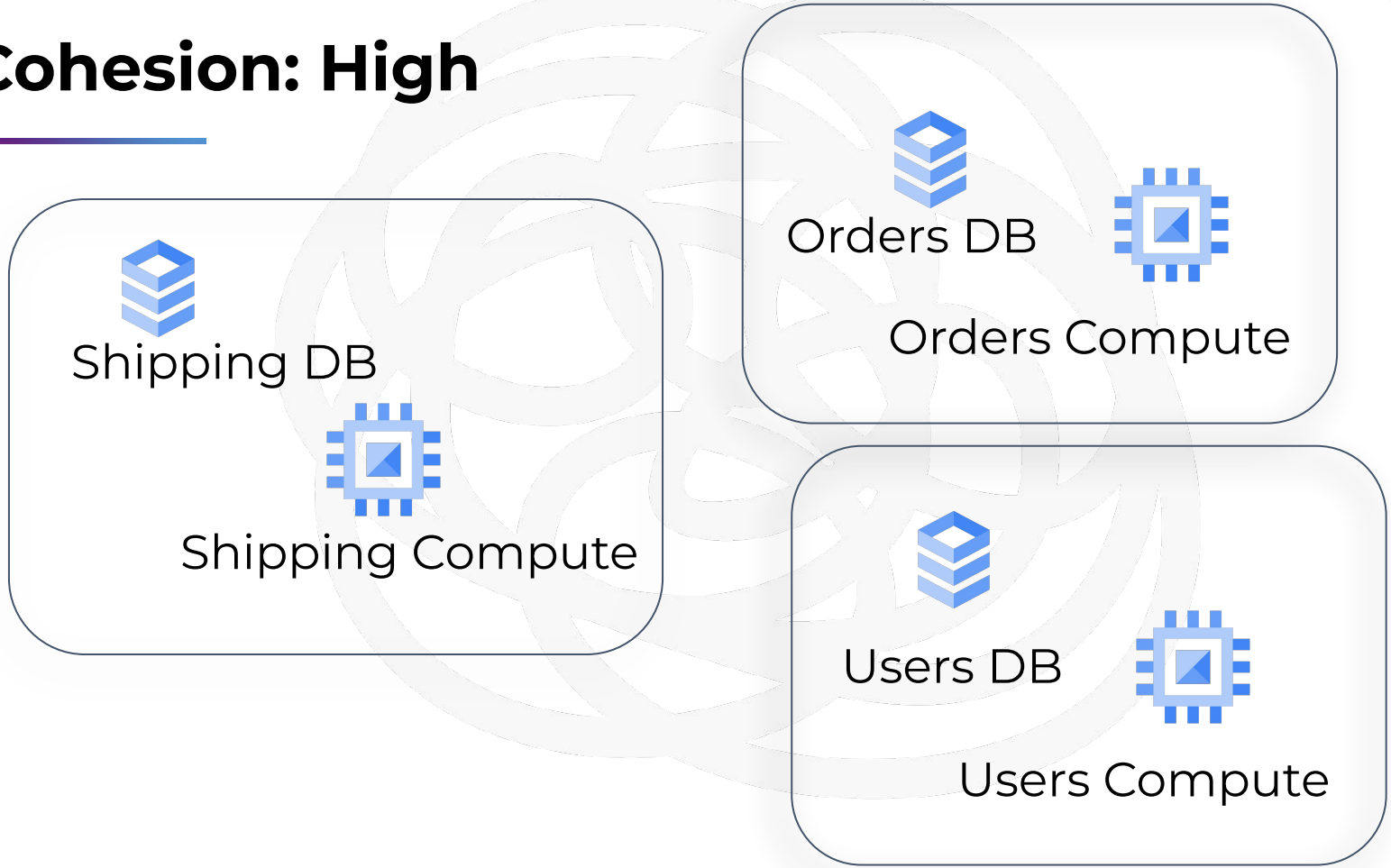
Cohesion: Low



Cohesion: High

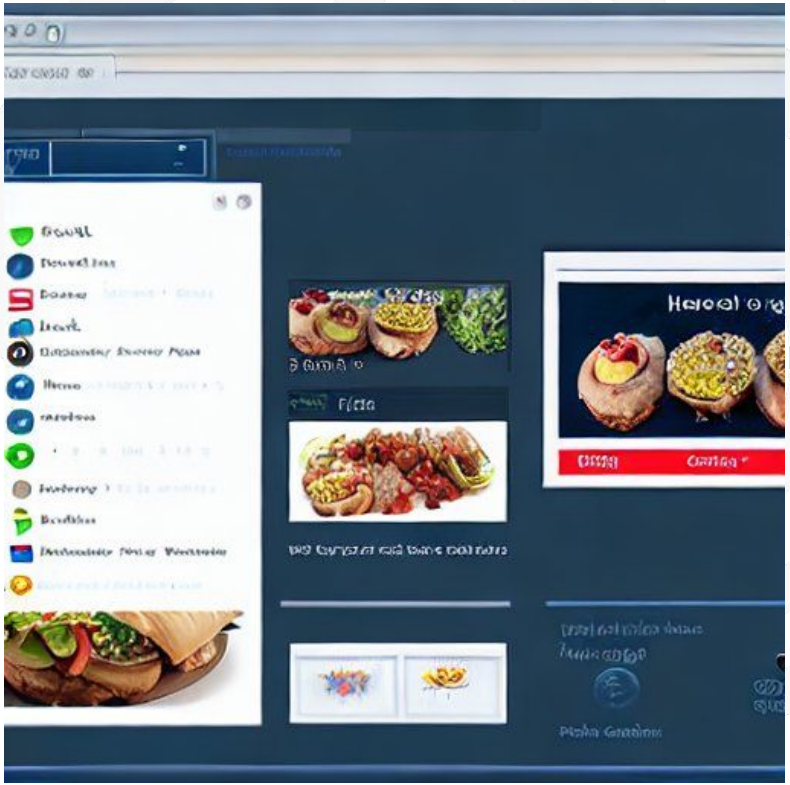


Cohesion: High

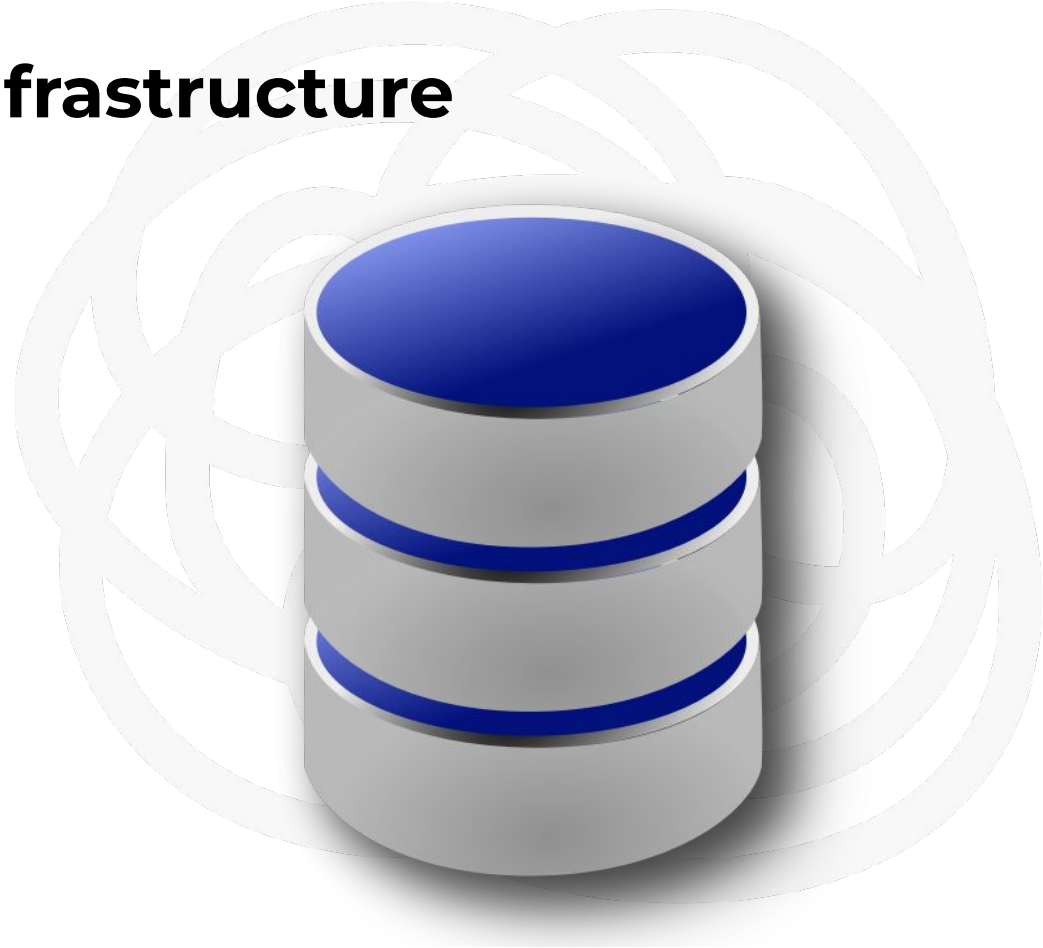


What should depend on what?
Stable vs. unstable systems

User-facing app changes often

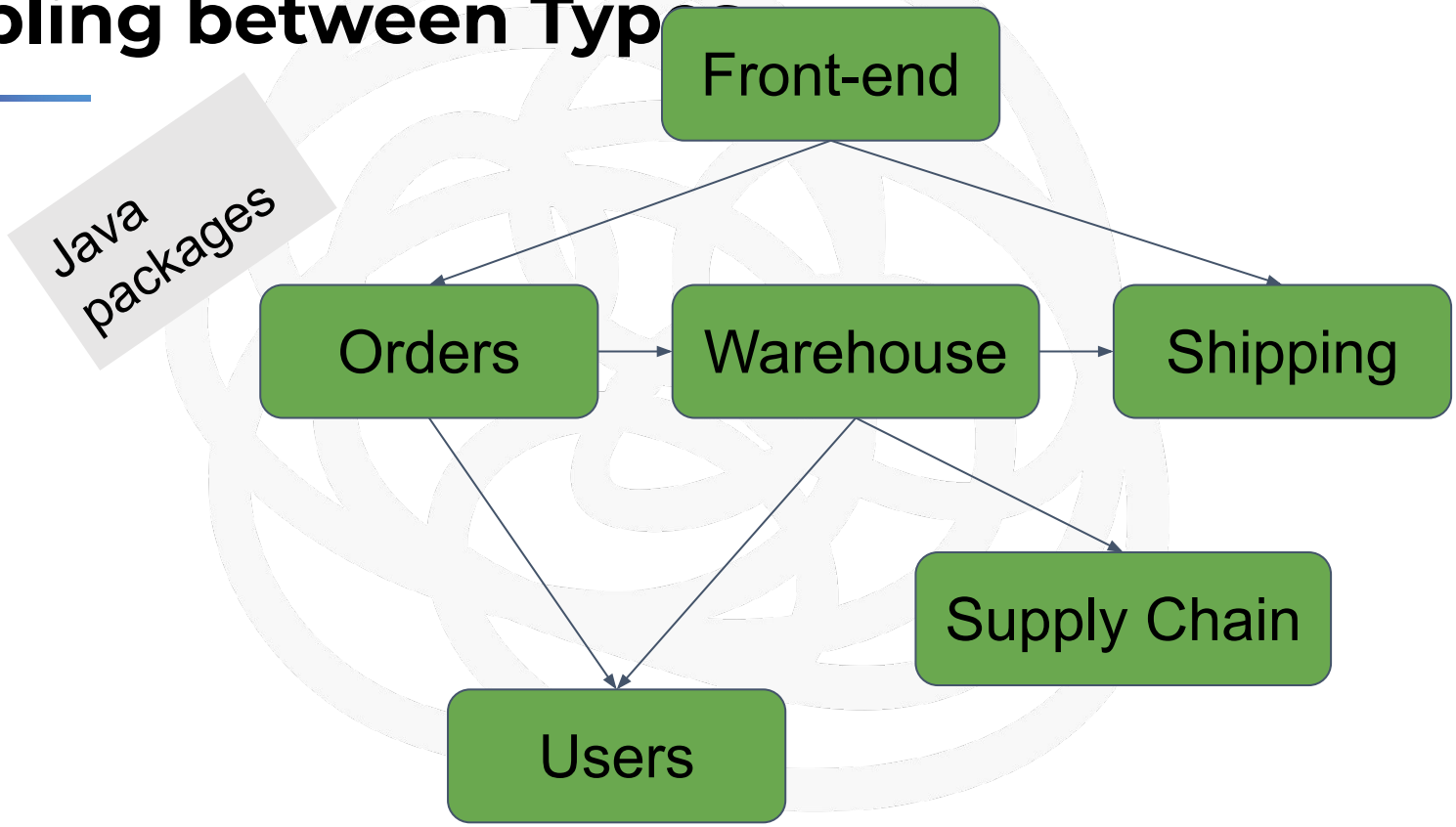


Stable infrastructure

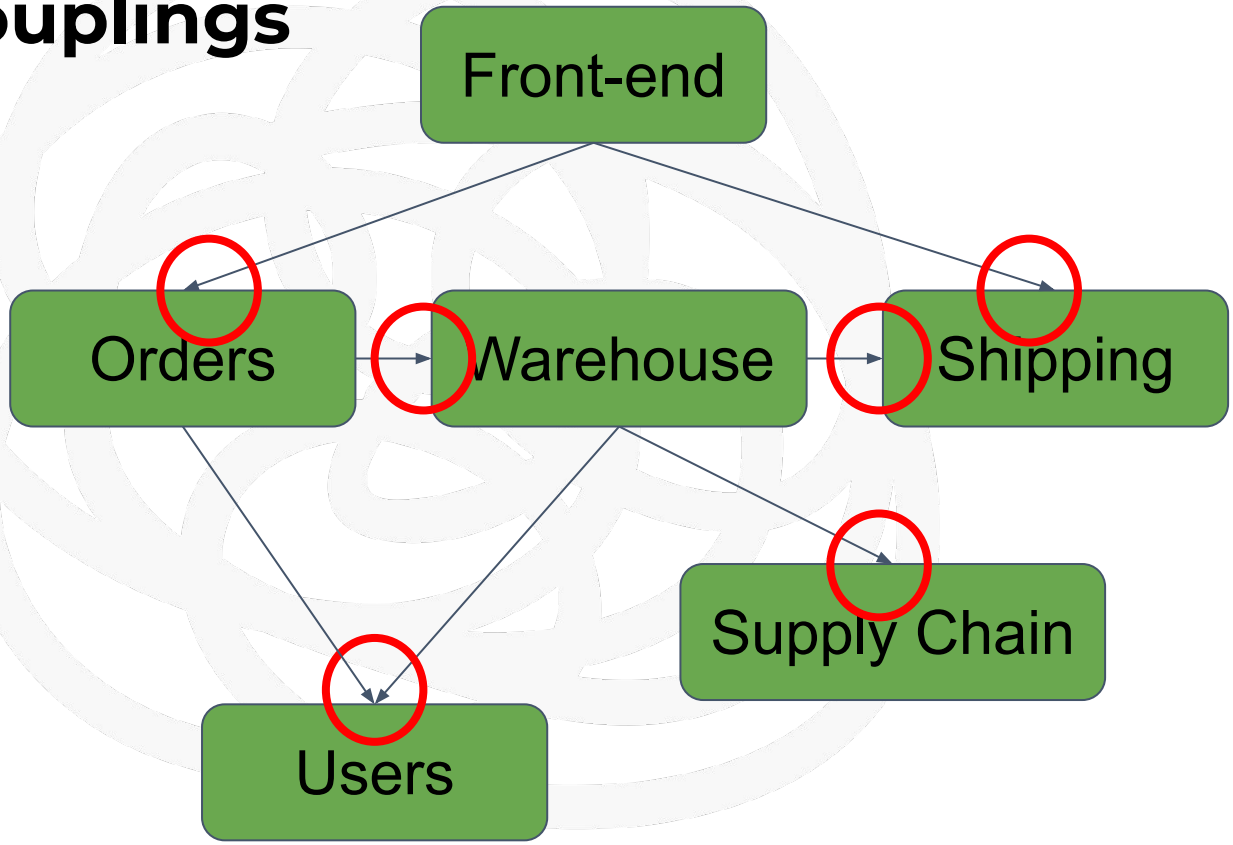


A nostalgic old tool
JDepend

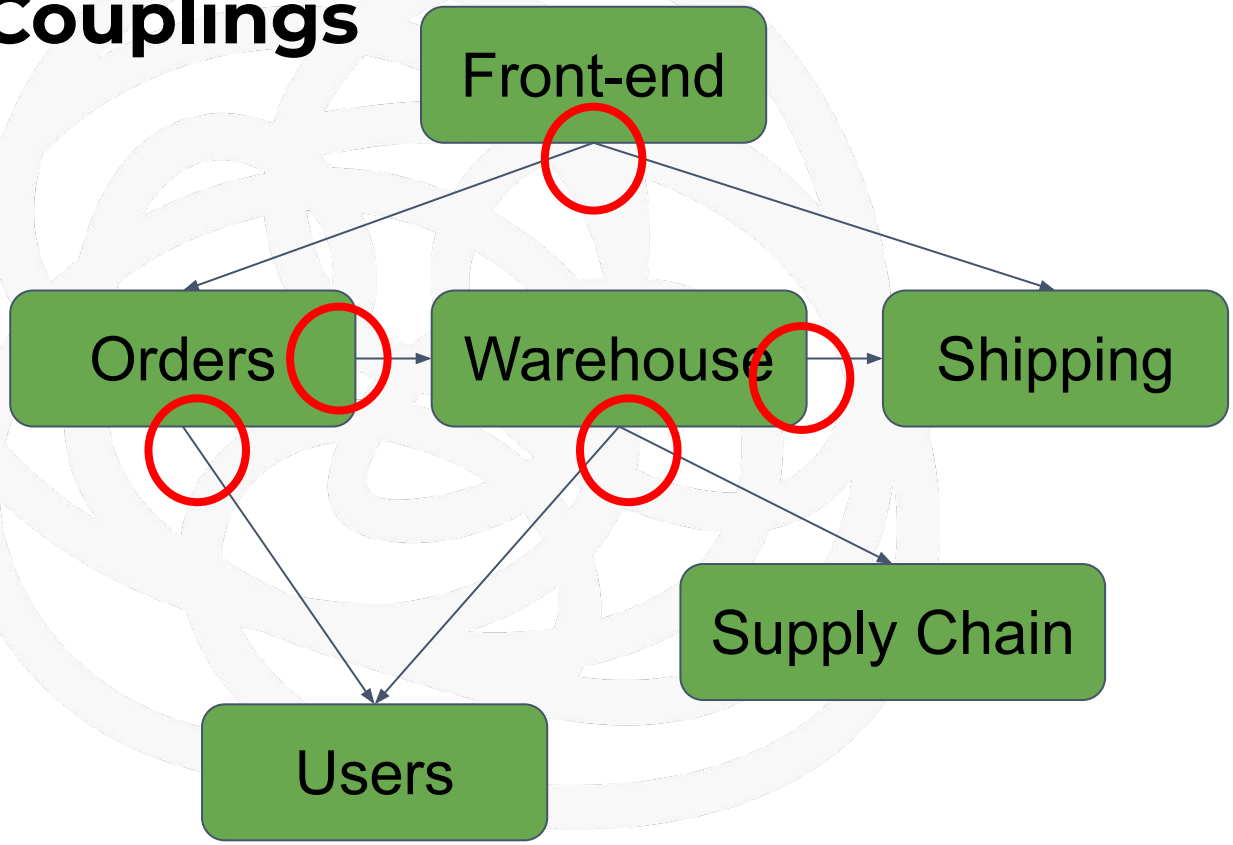
Coupling between Typ



Inbound Couplings



Outbound Couplings



Instability

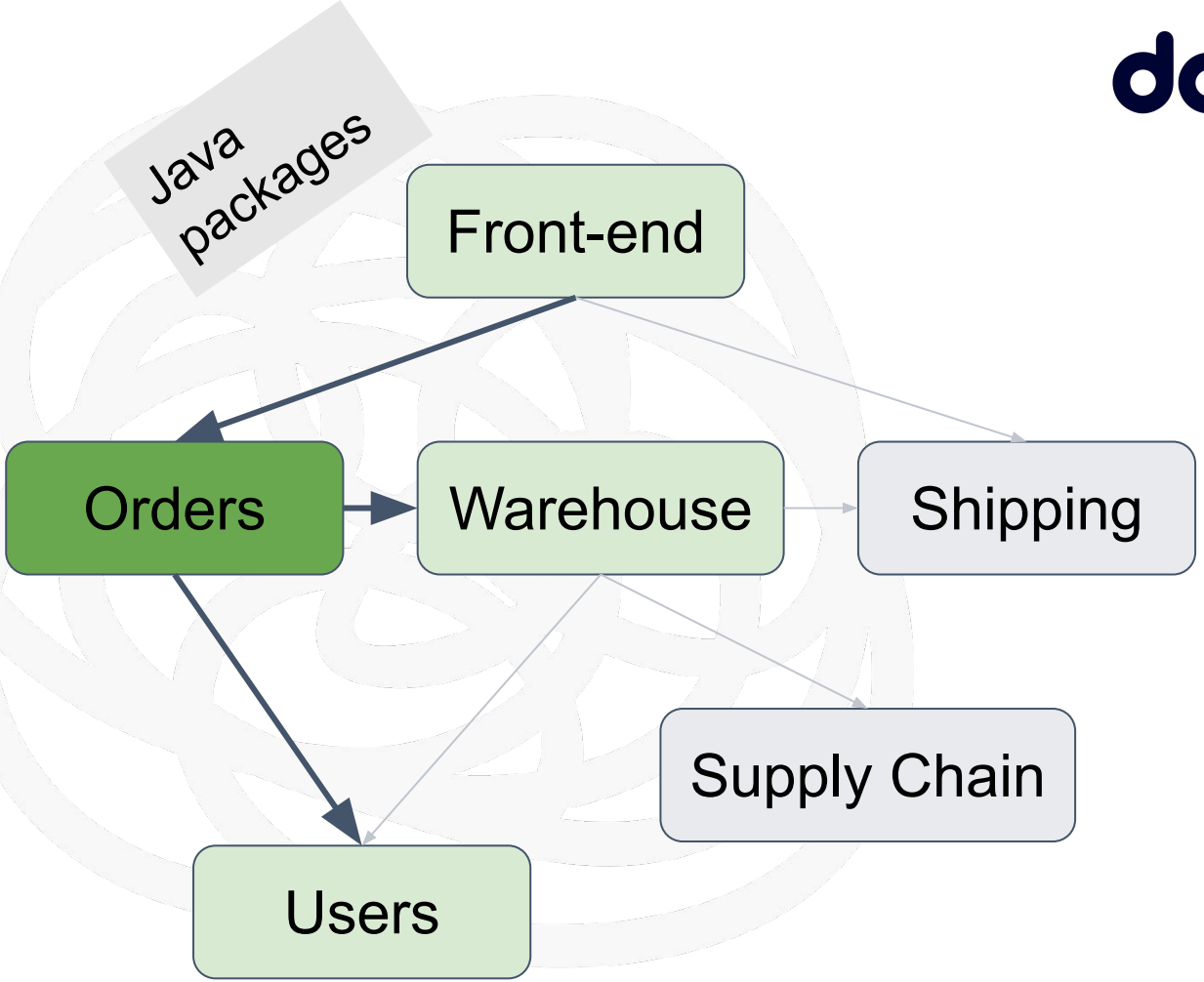
$$\frac{\text{Out arrows}}{\text{Total arrows}}$$

Instability

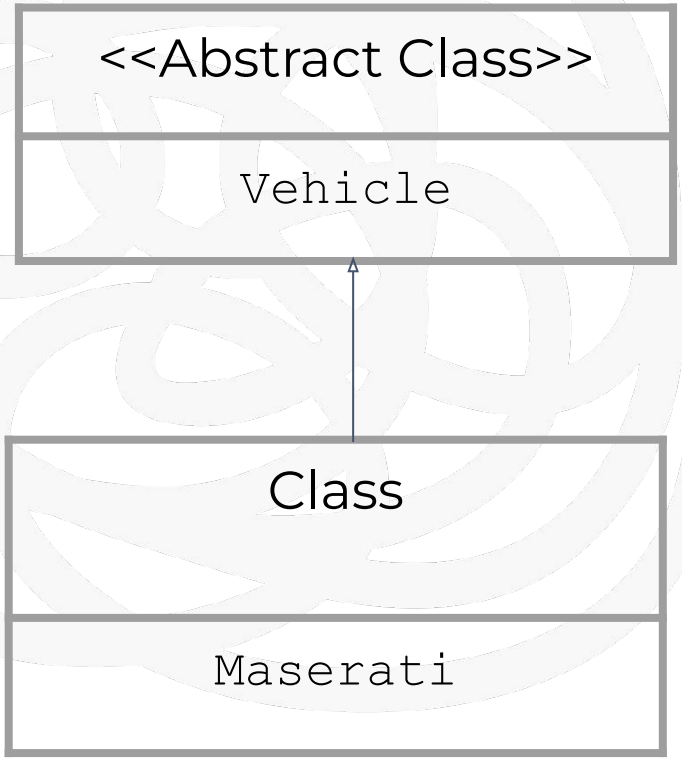
For “Orders”:

2 out, 1 in

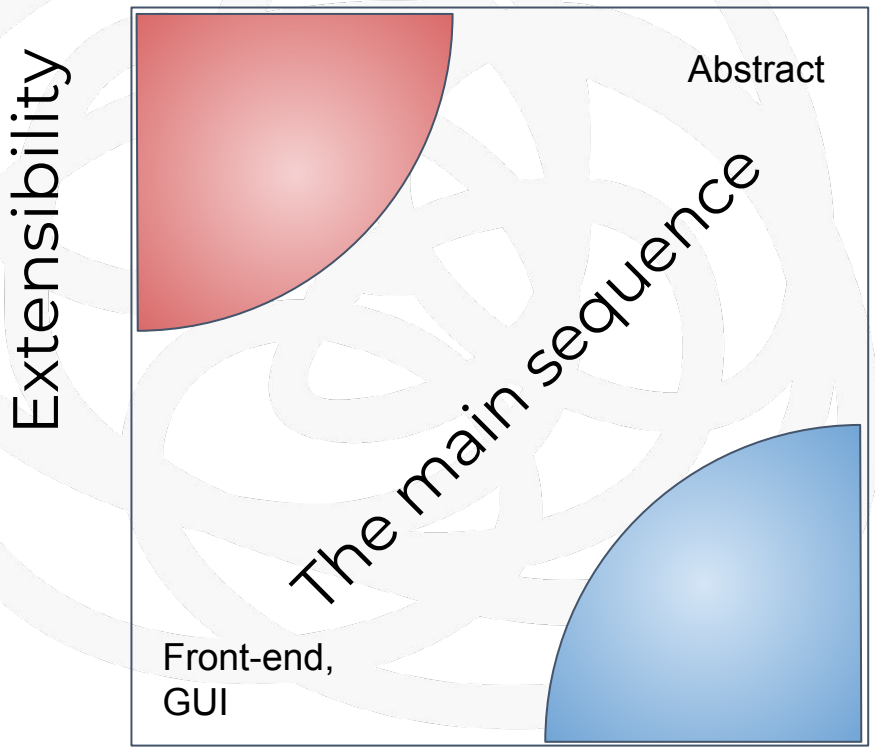
So, 3 total,
And so,
“Instability”=
out/total =
0.667



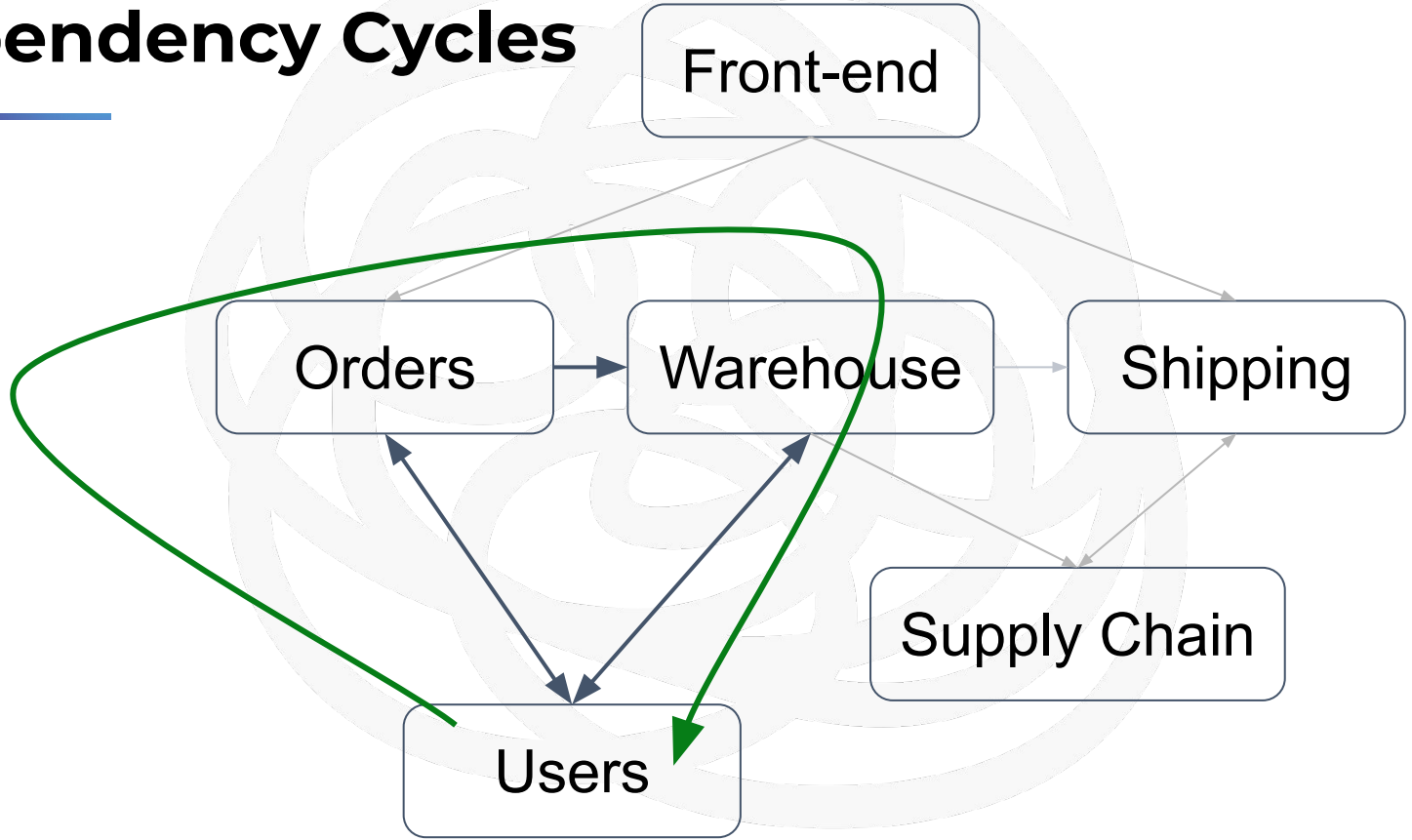
Abstractness



Distance from Main Sequence



Dependency Cycles



Example: `commons.imaging` “base” package

org.apache.commons.imaging

Afferent Couplings	Efferent Couplings	Instability
34	25	42.0%

Example: `commons.imaging.jpeg`

org.apache.commons.imaging.formats.jpeg

Afferent Couplings	Efferent Couplings	Instability
7	21	75.0%

Cycles

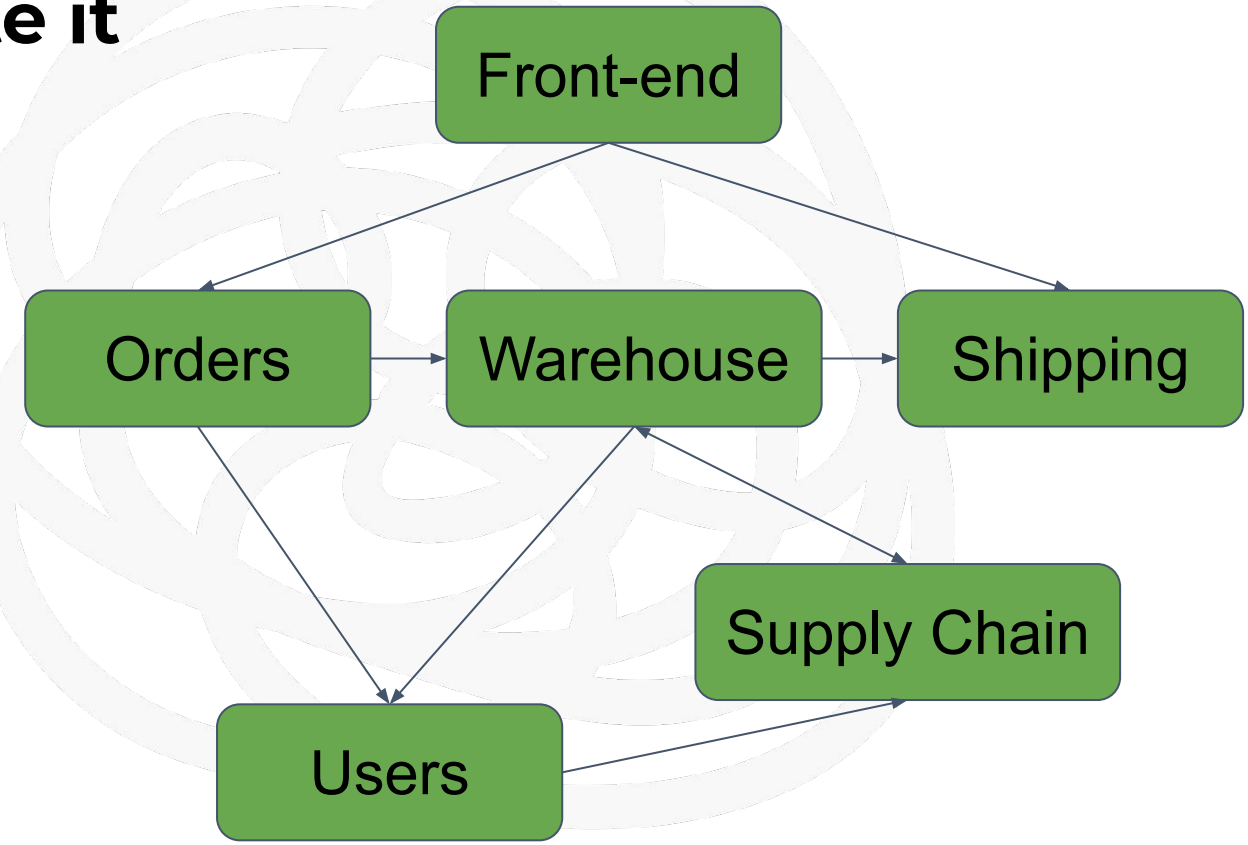
[summary] [packages] [cycles] [explanations]

Package	Package Dependencies
org.apache.commons.io	org.apache.commons.io.filefilter org.apache.commons.io

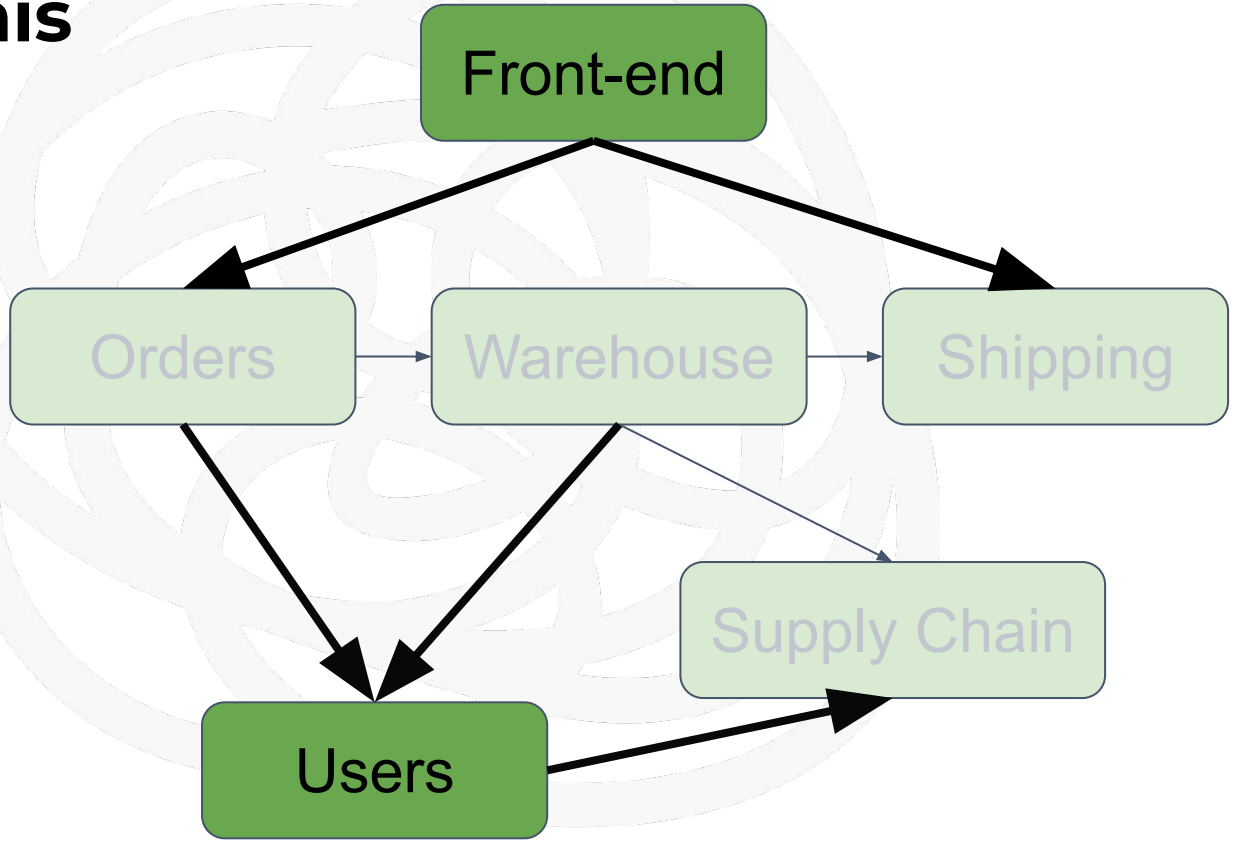
A new tool for the Cloud

Ferent

Why I wrote it



Focus on this



Ferent output

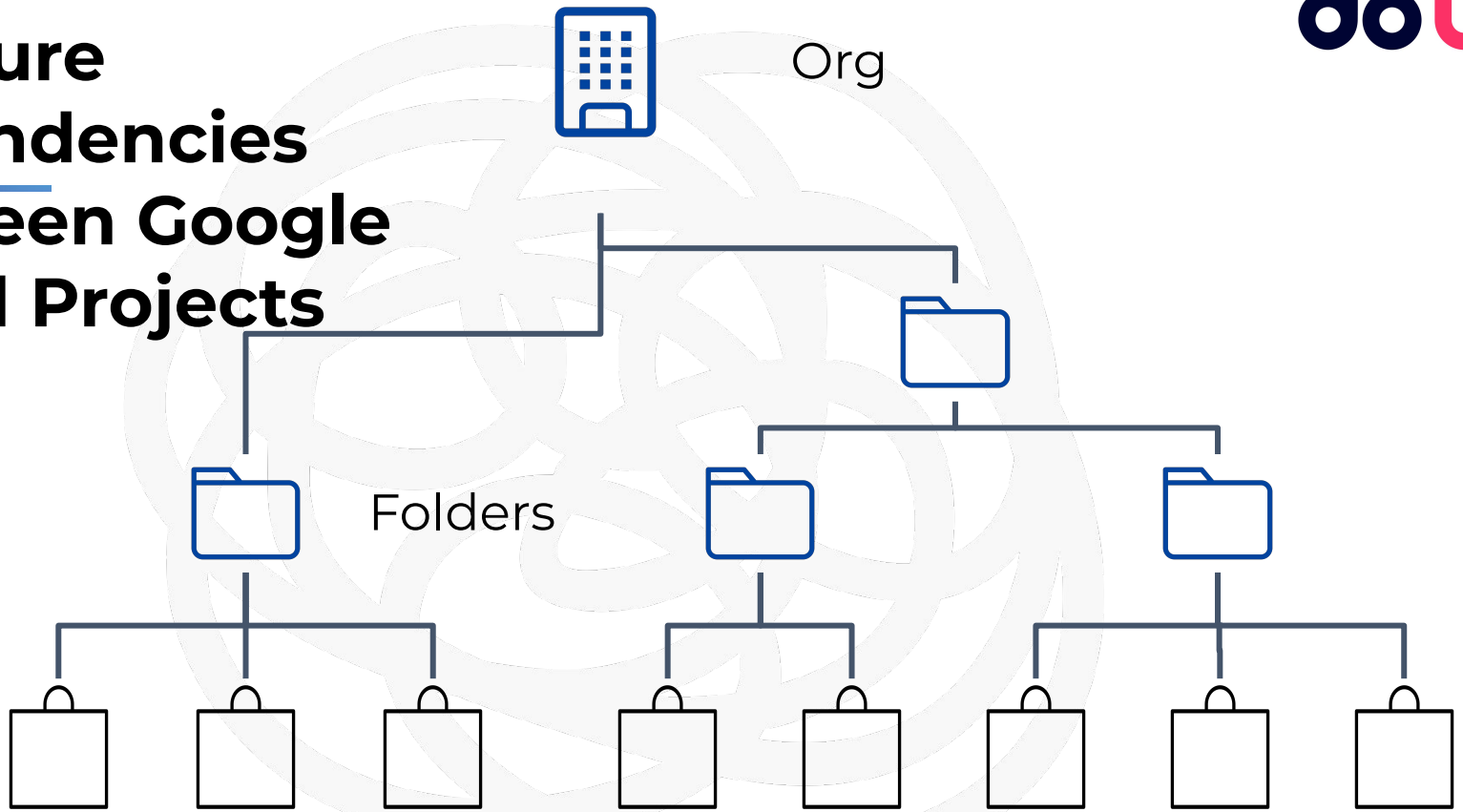
```
{"frontend"  {:arrow-in 0, :arrow-out 2,
              :instability 1.0},
{"users"  {:arrow-in 2, :arrow-out 1,
              :instability 0.33},
 //. . .
 :project-count 44,
 :cycles (["users" "warehouse"
            "supply-chain"],... )
 }
```

Unstable: ✔

Pretty Stable: ✔

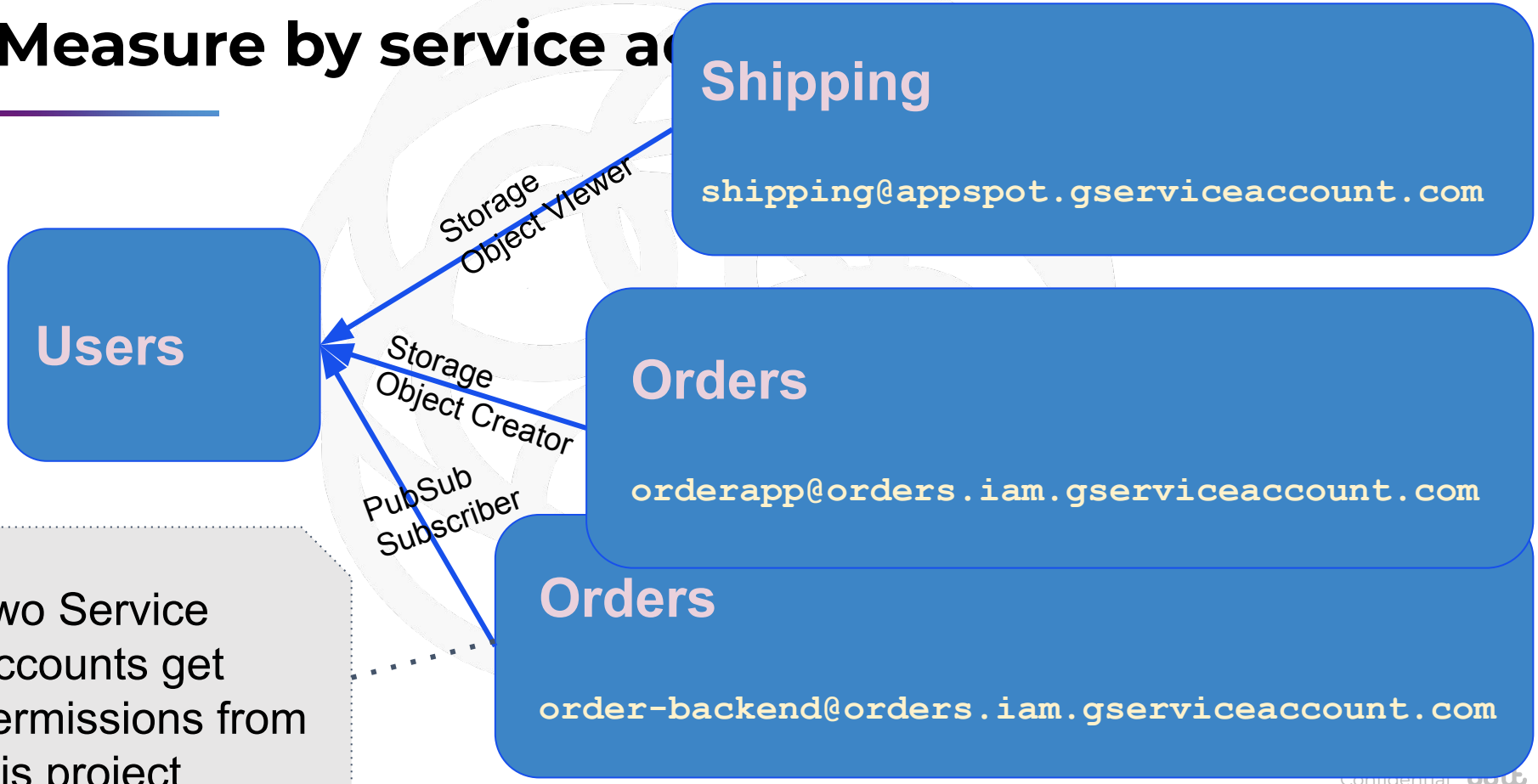
Oh no! Cycle

Measure dependencies between Google Cloud Projects



Projects

Measure by service a



Two Service Accounts get permissions from this project

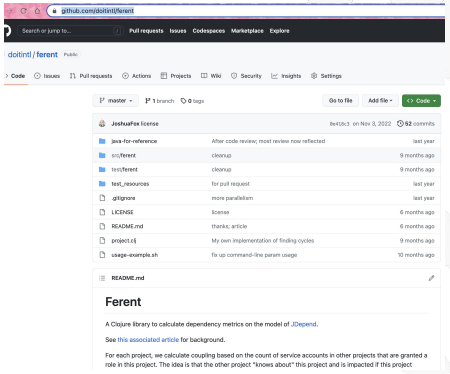
Fair Assumption?

Best practices

- **Internal** integrations are authenticated
- Use service accounts
- Preferred: Service account per app

Ferent Git repo

https://github.com/doitintl/ferent



I 💖 Clojure

```
(defn- depth-first-search [node path adjacency-map cycles]
 (cond
   (= node (first path))
   [(rotate-to-lowest path)]
   (some #{node} path)
   []
   :else
   (concat cycles
           (filter not-empty
                   (mapcat
                     (fn [child] (depth-first-search child (
                        conj path node) adjacency-map cycles))
                     (get adjacency-map node []))))))
```

Java Integration

```
(let [^SearchProjectsResponse search-response
    (..  svc (projects) (search)
    (setQuery filter) (setPageToken page-token

)...
```

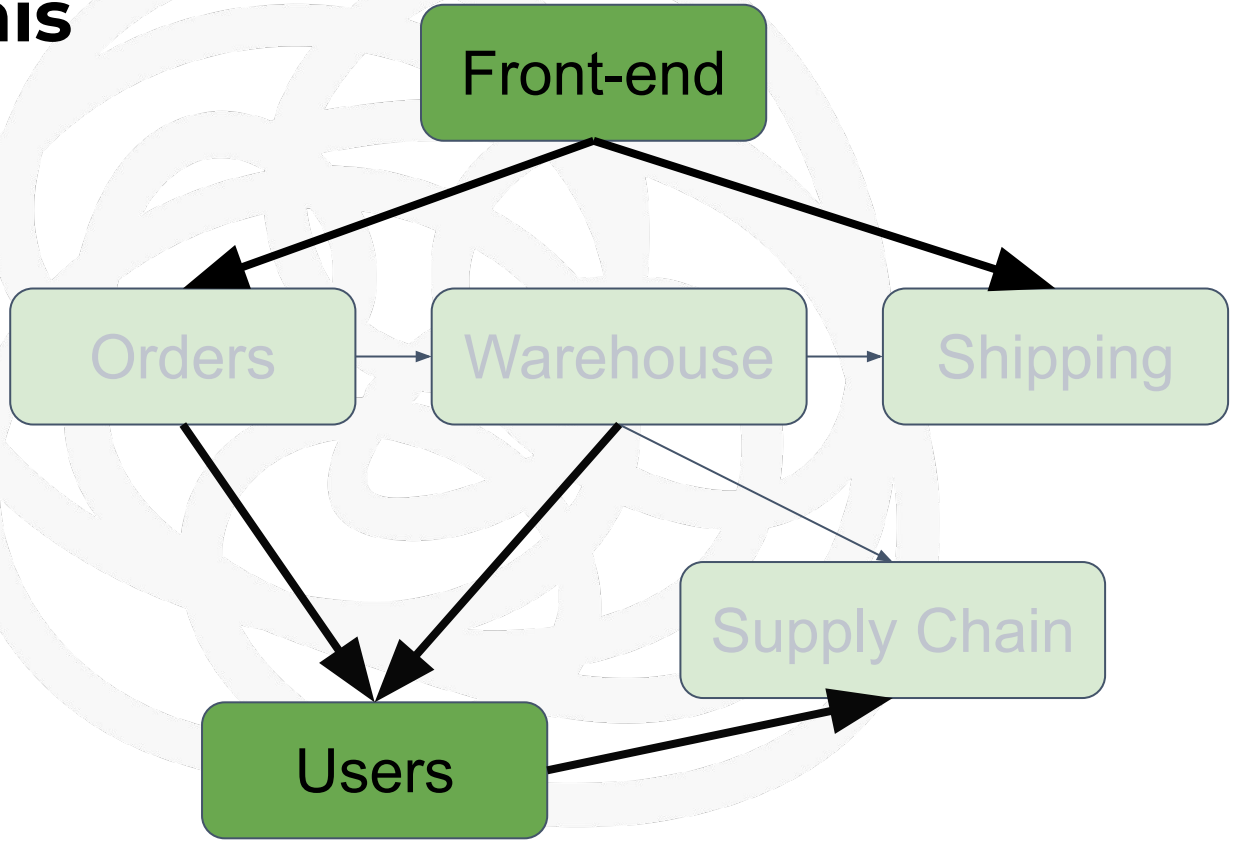
Commandline Integration

```
:require [babashka...
(process (conj '[gcloud projects get-ancestors]
proj-id))
```

How to run it

```
lein run --org-id 1234567890
-q "NOT displayName=doit* AND
NOT projectId=sys-*"
```

Focus on this



Ferent output

```
{"frontend"  {:arrow-in 0, :arrow-out 2,
              :instability 1.0},
{"users"  {:arrow-in 2, :arrow-out 1,
              :instability 0.33},
 //. . .
 :project-count 44,
 :cycles (["users" "warehouse"
            "supply-chain"],...)
 }
```

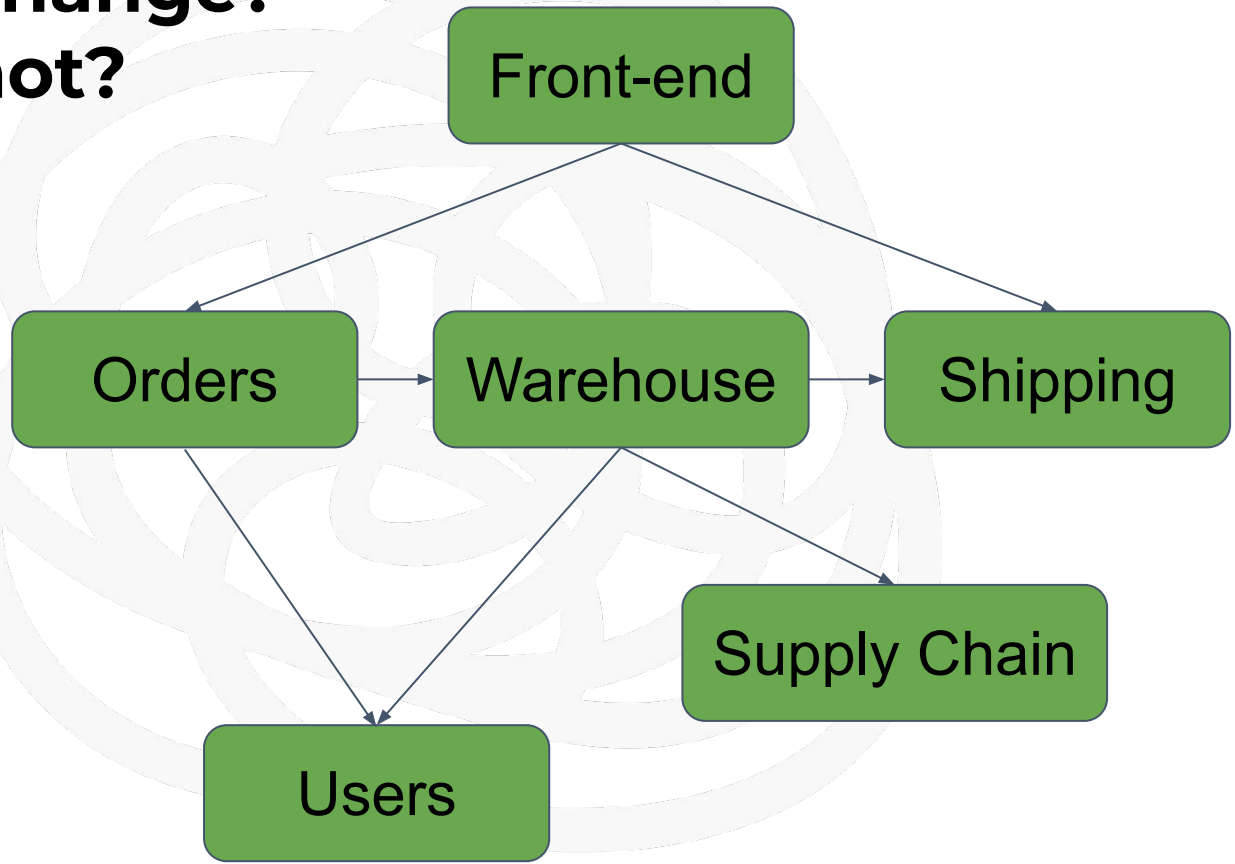
Unstable: ✔

Pretty Stable: ✔

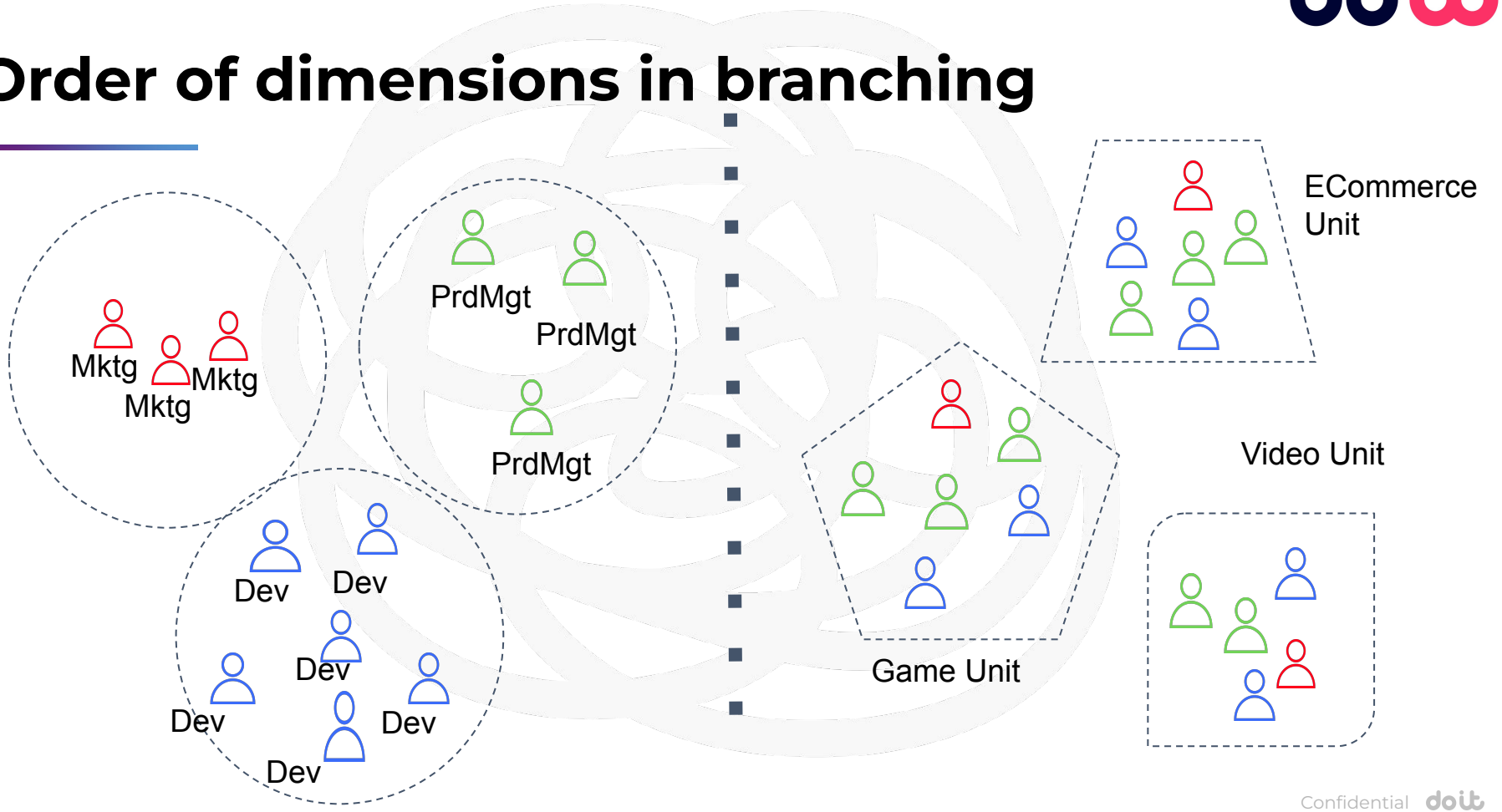
Oh no! Cycle

Key points
Take away concepts

What will change? What will not?

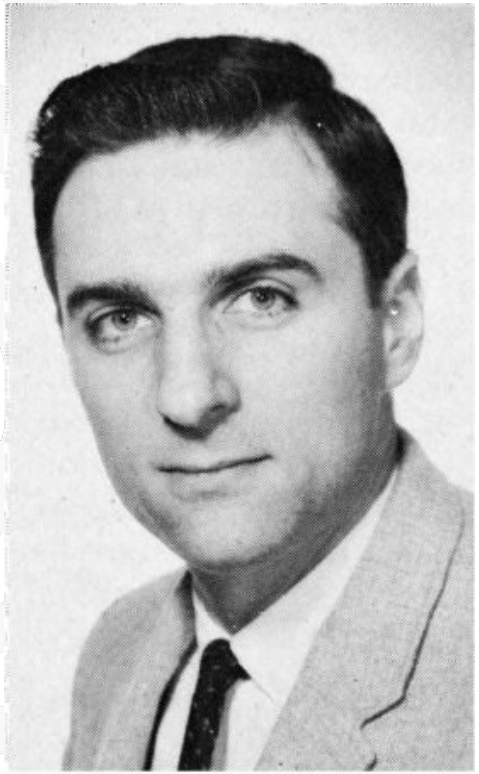


Order of dimensions in branching

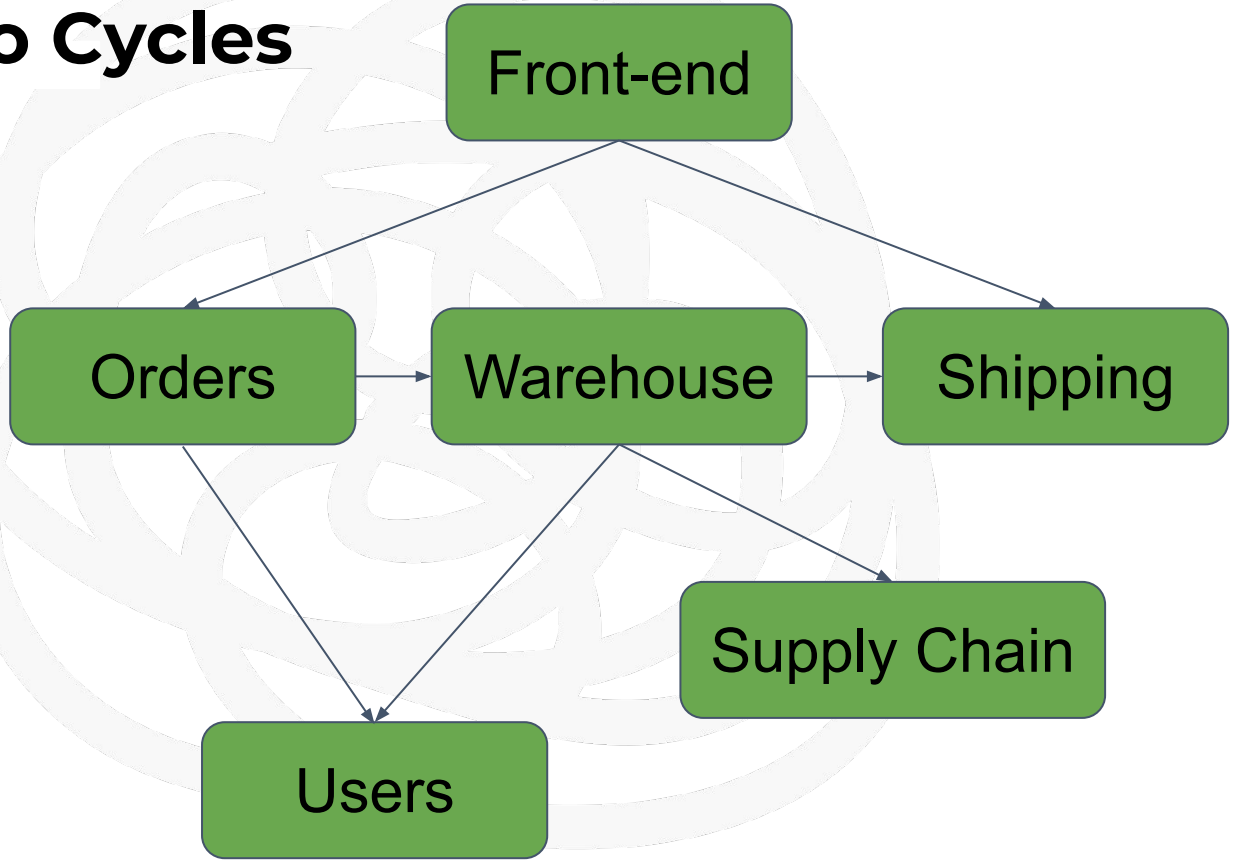


Conway's Law

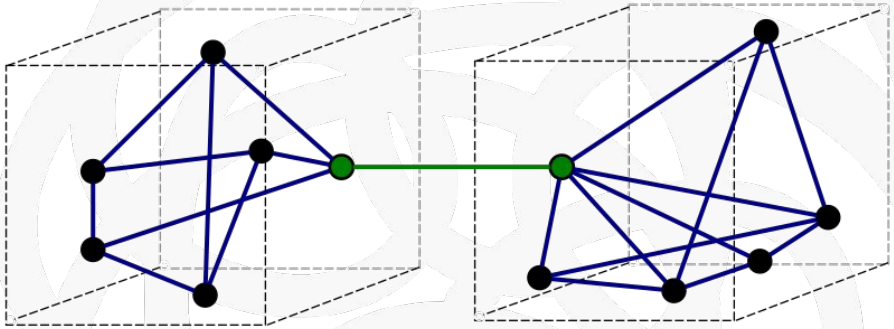
"Organizations which design systems are constrained to produce designs which are copies of the communication structures of these organizations."



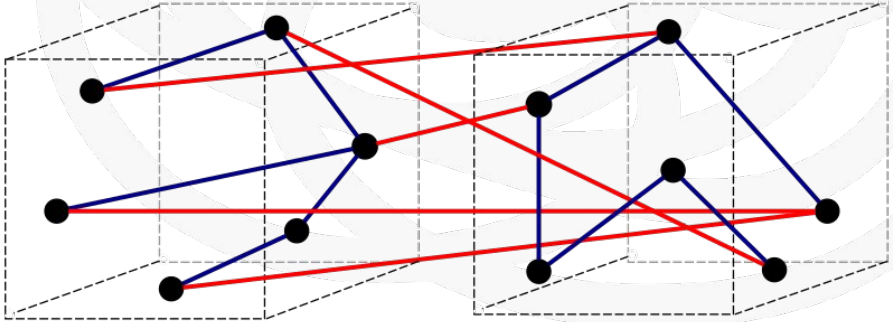
DAGged: No Cycles



Low Coupling, High Cohesion



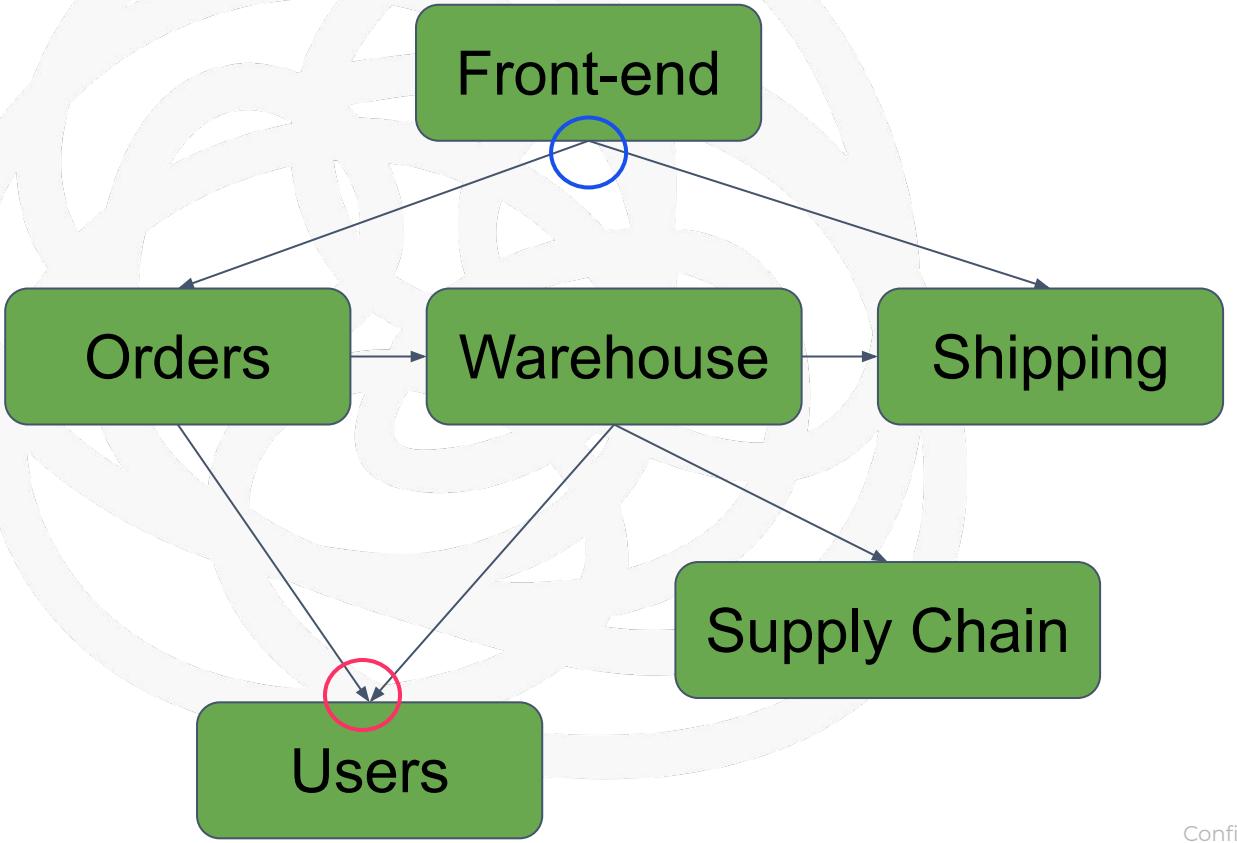
a) Good (loose coupling, high cohesion)



b) Bad (high coupling, low cohesion)

Wikipedia

Stability correlates to fraction inbound dependencies



Questions?

Please be in touch

joshua@doit-intl.com

doit

bit.ly/untangle-cloud