CPSC 3720
Lesson 8

Project Kickoff

Teaming

Microservices Start

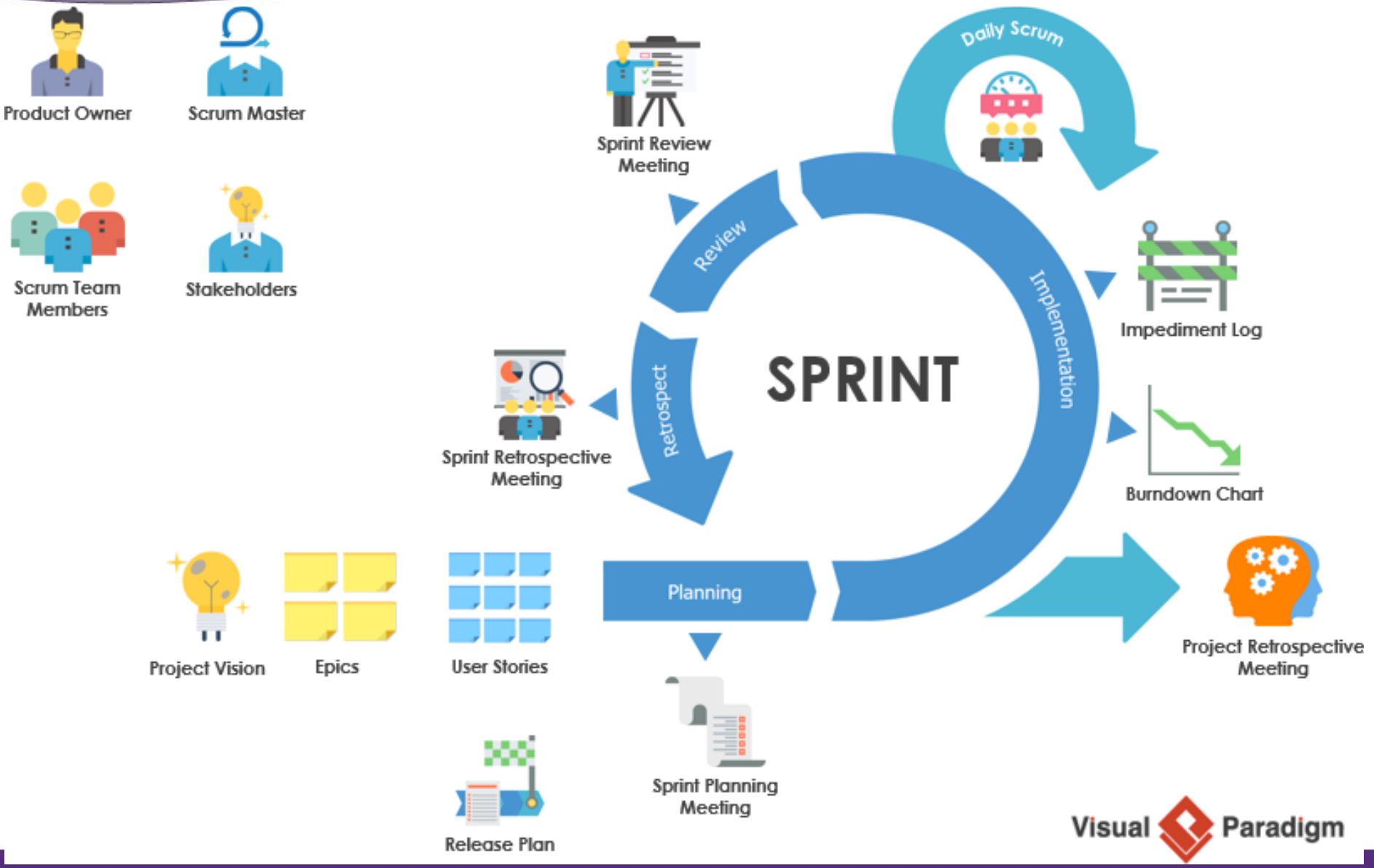
Connie Taylor
Professor of Practice

School of COMPUTING

Today's Objective

- **Project Kickoff**
- Importance of **Teams** in Software Engineering
- Begin **Microservices**
  - High-level understanding of Microservice-based architectures as they are becoming the norm in software architectures today
  - Relate how this architecture is relevant to our class project

Scrum in 1 Picture



User "Roles"

- A **User Role** is a collection of defining attributes that characterize a population of users and their intended interactions with the system.

Source: *Software for Use* by Constantine and Lockwood (1999).

User Stories

User stories are short, simple descriptions of a feature told from the perspective of the person who desires the new capability, usually a user or customer of the system. They typically follow a simple template:

As a < type of user/user role > (WHO), I want < some goal > (WHAT), so that < some reason > (WHY)

The Three C's of User Stories

Card

- Stories are traditionally written on note cards.
- Cards may be annotated with estimates, notes, etc.

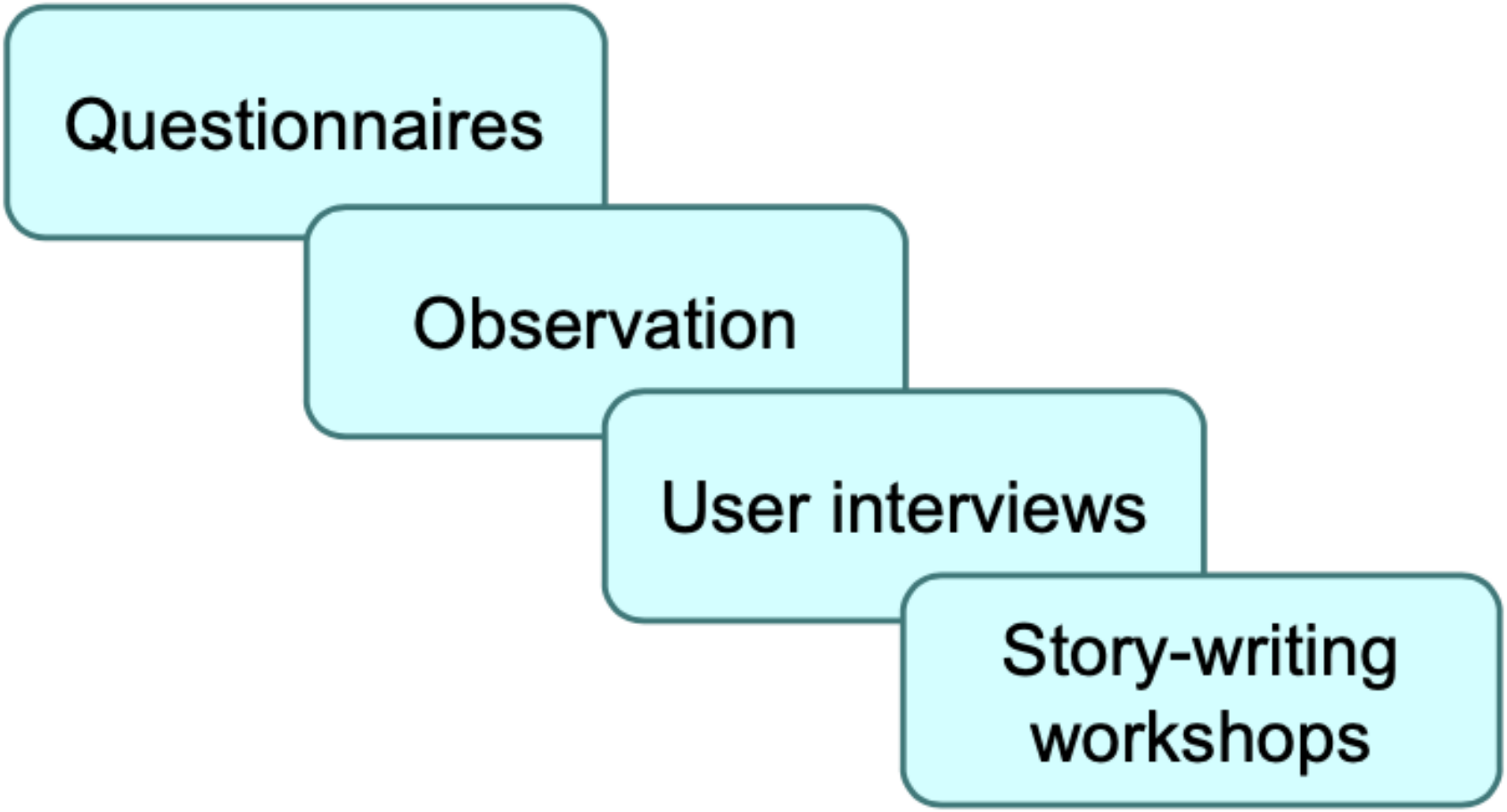
Conversation

- Details behind the story come out during conversation with customer

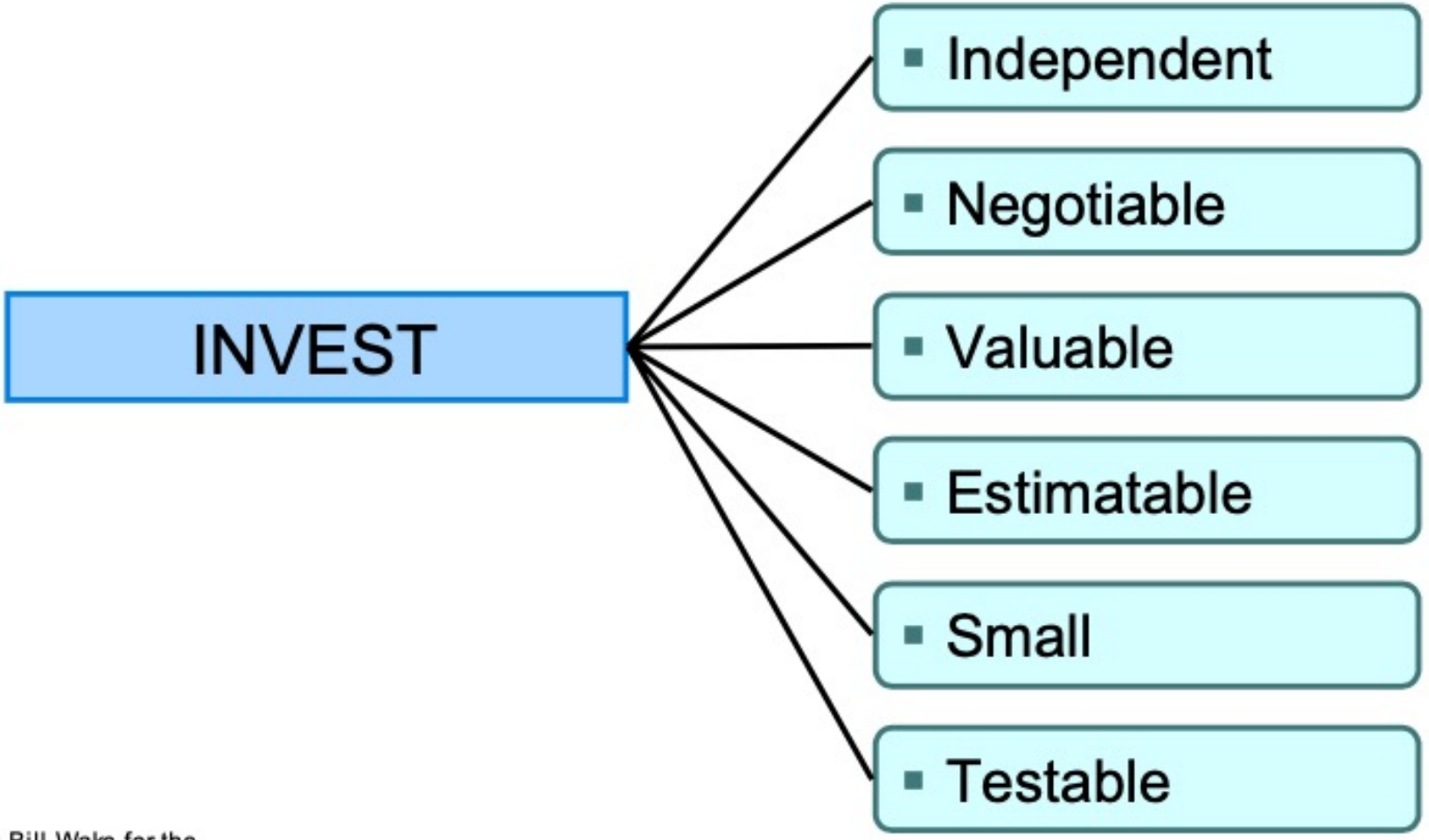
Confirmation

- Acceptance tests confirm the story was coded correctly

Techniques for Gathering Stories



What makes a good user story?



Thanks to Bill Wake for the acronym. See www.xp123.com.

User Stories – General Guidelines

INVEST

S.A.M. – helps with testable

- Specific
- Attainable
- Measurable

Story Hierarchy

- Product – the largest chunk of value
- Epic – a piece of value that is larger than a story but smaller than the whole product
- User Story – a small piece of value that can be implemented in at most a week or two; **broken down into tasks for each sprint**

Stories should answer- Who, What, Why?

- Who – who the user story is for (As a User)
- What – the functionality that the user story implements (I want)
- Why – the reason the user needs the user story (so that)

TigerChow Project

The team project for Spring 2024 is to create services and APIs as required by the TigerChow food delivery online application. The teams will develop services and their corresponding APIs to support a set of epics as specified by the Product Owner (Prof. Taylor). Sprint management will be done in Trello. The API testing will be done in Postman using an API application built and deployed in AWS.

Technologies:

- Open API Specification
- JSON
- Postman
- Trello
- AWS: API Gateway, Lambda, Node, DynamoDB

Example APIs:

Shopify APIs

TigerChow Project Sprints

- Project is 325 points (32.5% of grade).
- All team members expected at sprint reviews and, if not, will impact the **absent** student's grade.
- There will be 5-6 total sprints.
- More details after the Kickoff sprint.

TigerChow Users/Roles Recommendation

As a < type of user/user role >, I want < some goal >, so that < some reason > (WHY)

Customers (role)

- On campus students
- Fraternities and sororities
- Off campus students
- Faculty
- Clemson locals
- Delivery drivers
- Restaurant managers and personnel
- Food critics
- Event planners
- People visiting the area
- Sports teams
- Loyalty program members
- Mobile users
- People who don't know technology
- Colorblind – visually impaired users
- Cash only customers
- 21 and up consumers

IGNORE

- Job applicants
- HR

Restaurant Staff (role)

- Fulfill orders

Delivery drivers (role)

- Delivery driver

TigerChow System Admin (role)

- IT users

Customer Service User (role)

- Customer Support staff to help customers with orders

Business User (role)

- Business users (run reports to manage business)
- Marketing
- HR
- TigerChow executive management
- Sales
- Partners/sponsors/advertisers (promotion on website)

TigerChow Project: Teams

- 17 teams total to keep teams “Agile-sized” (4 people per team)
- Team members have been set up as Trello Workspaces and Canvas groups for grading
- Be sure to make sure you have access to Trello (see today’s module for link to your team workspace)

TigerChow Project – Sprint 0

Sprint 0 (total of 25 points):

- **Team Kickoff and TigerChow epics** using the Trello Board instructions.
- COPY the Kickoff and Epic boards to your workspace
- 2-3 minute review per team next Tuesday 2/13 to present:
  - Team Kickoff Board with team name and logo
  - Epic and Services board
- 10 points for thoughtful kickoff board/logo/name
- 10 points for thoughtful Epics and Services
- 5 points for the Sprint Review (if you are not present you lose the points)
- Team Survey due at end of day Wed 2/14 (if you don't do the survey, it will impact your grade!)

The Teamwork Survey will impact your grade!

At the end of each sprint, your survey score will impact your grade as follows:

0 >= survey score < 2 for a Sprint – 0 for that sprint
2 >= survey score < 2.5 for a Sprint – 70% of team Sprint grades
2.5 >= survey score < 3 for a Sprint – 80% of team Sprint grades
3 >= survey score < 3.5 for a Sprint – 90% of team Sprint grades

If the entire team agrees that you were scored unfairly for a particular sprint (email sent copying each member of the team) - I will reconsider.

The Teamwork Survey will impact your grade!

TEAMWORK VALUE RUBRIC

for more information, please contact value@aacu.org

Definition

Teamwork is behaviors under the control of individual team members (effort they put into team tasks, their manner of interacting with others on team, and the quantity and quality of contributions they make to team discussions.)

Evaluators are encouraged to assign a zero to any work sample or collection of work that does not meet benchmark (cell one) level performance.

	Capstone 4	Milestones 3	Milestones 2	Benchmark 1
Contributes to team meetings	Helps the team move forward by articulating the merits of alternative ideas or proposals.	Offers alternative solutions or courses of action that build on the ideas of others.	Offers new suggestions to advance the work of the group.	Shares ideas but does not advance the work of the group.
Facilitates the contributions of team members	Engages team members in ways that facilitate their contributions to meetings by both constructively building upon or synthesizing the contributions of others as well as noticing when someone is not participating and inviting them to engage.	Engages team members in ways that facilitate their contributions to meetings by constructively building upon or synthesizing the contributions of others.	Engages team members in ways that facilitate their contributions to meetings by restating the views of other team members and/or asking questions for clarification.	Engages team members by taking turns and listening to others without interrupting.
Individual contributions outside of team meetings	Completes all assigned tasks by deadline; work accomplished is thorough, comprehensive and advances the project. Proactively helps other team members complete their assigned tasks to a similar level of excellence.	Completes all assigned tasks by deadline; work accomplished is thorough, comprehensive and advances the project.	Completes all assigned tasks by deadline; work accomplished advances the project.	Completes all assigned tasks by deadline.
Fosters constructive team climate	Supports a constructive team climate by doing all of the following: • Treats team members respectfully by being polite and constructive in communication. • Uses positive vocal or written tone, facial expressions, and/or body language to convey a positive attitude about the team and its work. • Motivates teammates by expressing confidence about the importance of the task and the team's ability to accomplish it. • Provides assistance and/or encouragement to team members.	Supports a constructive team climate by doing any three of the following: • Treats team members respectfully by being polite and constructive in communication. • Uses positive vocal or written tone, facial expressions, and/or body language to convey a positive attitude about the team and its work. • Motivates teammates by expressing confidence about the importance of the task and the team's ability to accomplish it. • Provides assistance and/or encouragement to team members.	Supports a constructive team climate by doing any two of the following: • Treats team members respectfully by being polite and constructive in communication. • Uses positive vocal or written tone, facial expressions, and/or body language to convey a positive attitude about the team and its work. • Motivates teammates by expressing confidence about the importance of the task and the team's ability to accomplish it. • Provides assistance and/or encouragement to team members.	Supports a constructive team climate by doing any one of the following: • Treats team members respectfully by being polite and constructive in communication. • Uses positive vocal or written tone, facial expressions, and/or body language to convey a positive attitude about the team and its work. • Motivates teammates by expressing confidence about the importance of the task and the team's ability to accomplish it. • Provides assistance and/or encouragement to team members.
Responds to conflict	Addresses destructive conflict directly and constructively, helping to manage/resolve it in a way that strengthens overall team cohesiveness and future effectiveness	Identifies and acknowledges conflict and stays engaged with it	Redirecting focus toward common ground, toward task at hand (away from conflict)	Passively accepts alternate viewpoints/ideas/opinions.

The Importance of Teams in Software Development

- **Conway's Law**
  - "Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure." — Melvin E. Conway

- **Team = Software**
  - "You can't have great software without a great team, and [many] software teams behave like dysfunctional families" - Jim McCarthy

Don’t Flip the Bozo Bit

https://YouTube/QSKLm8E6KyE

Dream Teams

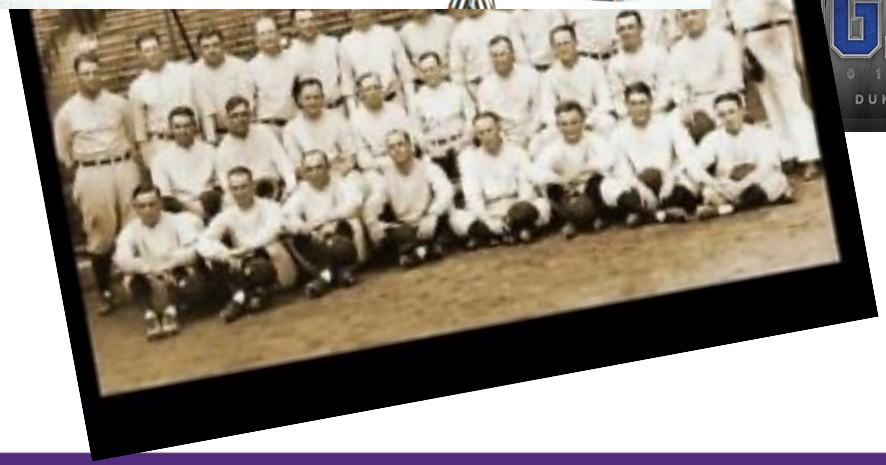

4: 10
24
BULLS
91
BULLS
33
BULLS
23
BULLS
9
BULLS
7


IRWIN BELK TRACK
FETZER FIELD


USA
9
15


DUKE
DUKE BASKETBALL


16
9
NATIONAL
CHAMPIONS
CLEMSON TIGERS
16
44




BRADY
12

Breakout

- Each person in the team share the best team you were part of (can be school, sports, hobby, etc.) and the key characteristics that made the team great

- Someone in the team keep track of these characteristics (note any that are repeats across the team)

- After ~5 minutes we will regroup as a class and someone from the team should report out your “great” team characteristics to the larger group

Microservices ?

Companies that moved to microservices

Airbnb

Netflix

Uber

LinkedIn

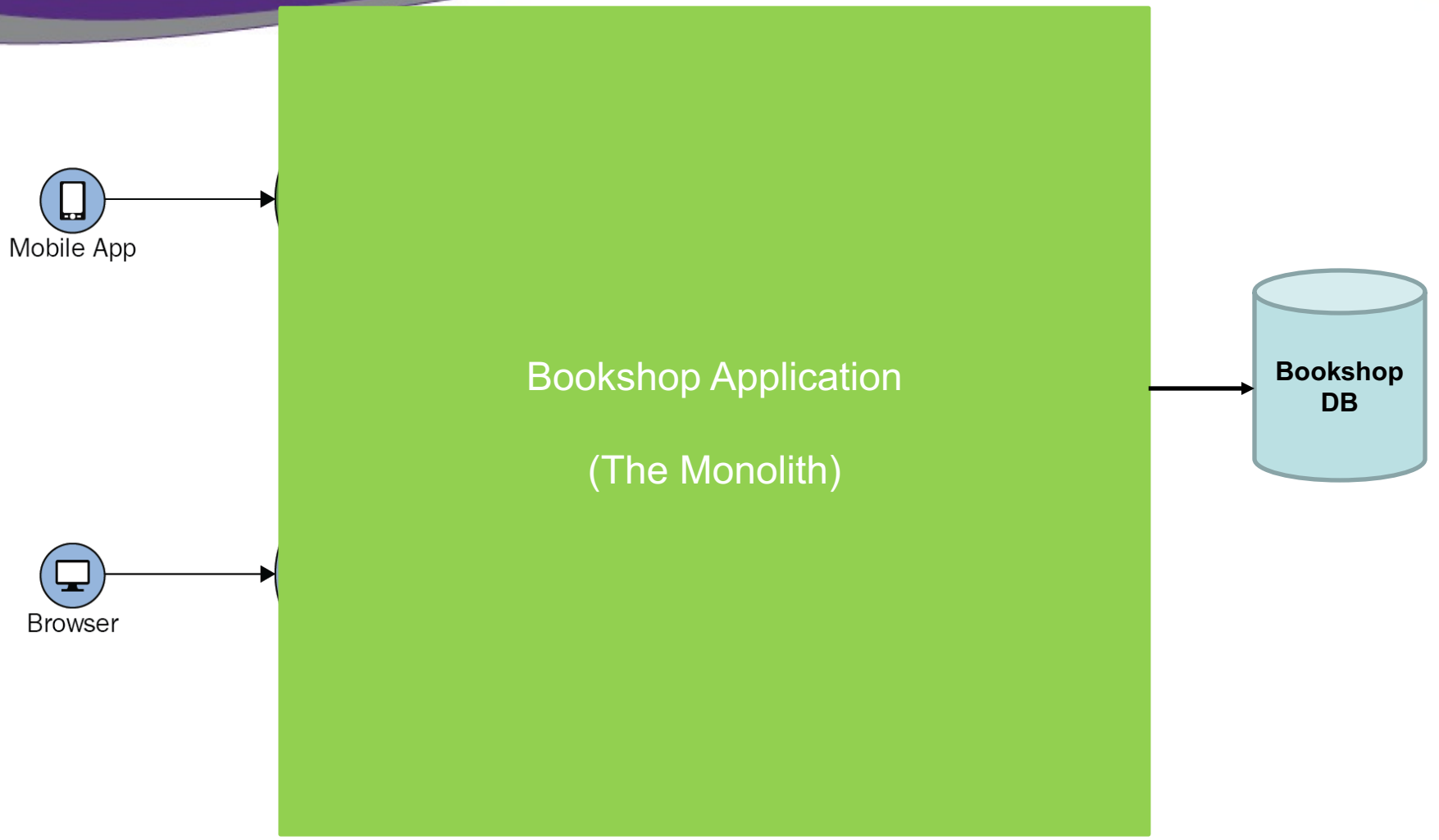
PayPal

The Guardian

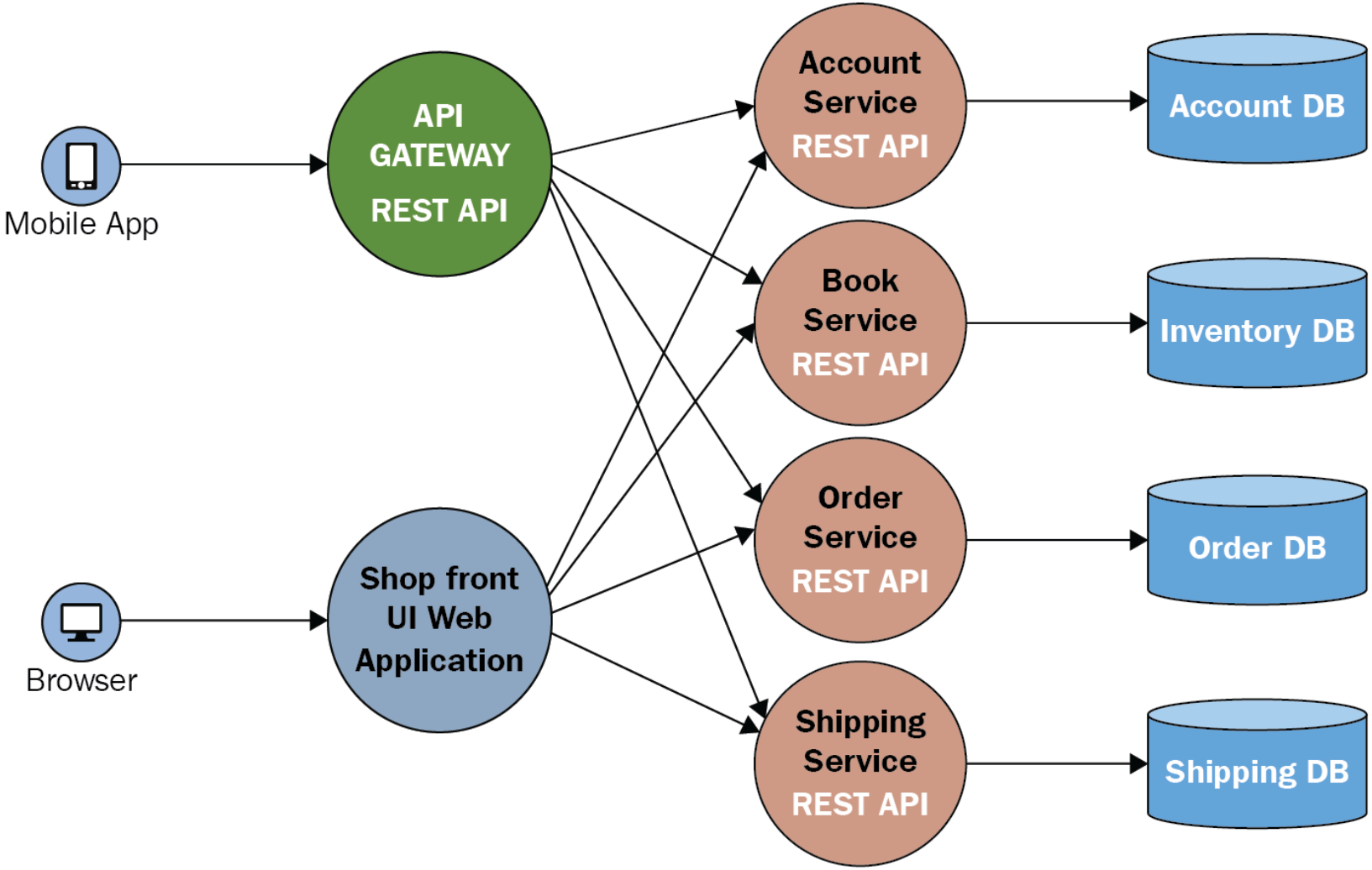
eBay

Amazon

Before Microservices : Online Bookshop



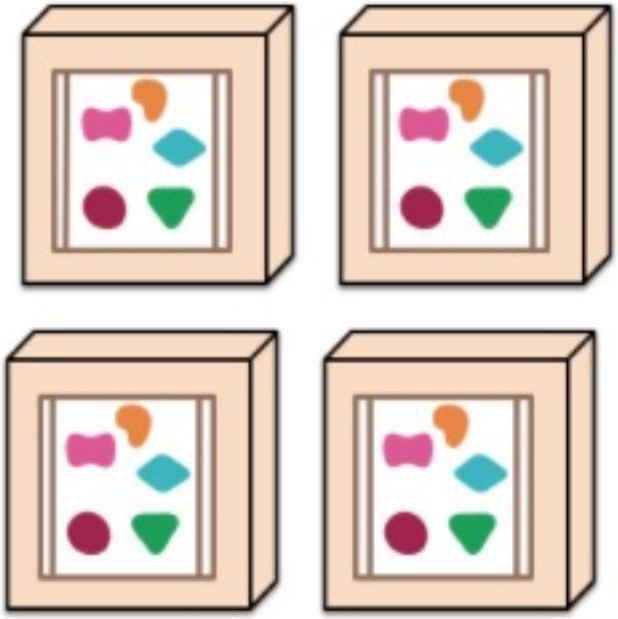
Microservices Example: Online Bookshop



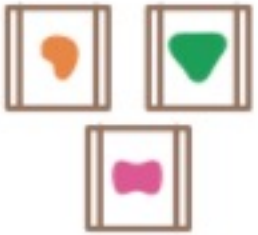
Applications: From Monoliths to Microservices

A monolithic application puts all its functionality into a single process...

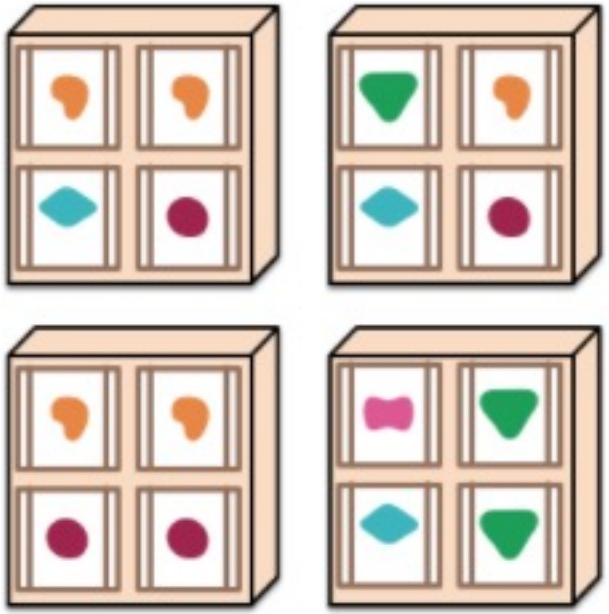
... and scales by replicating the monolith on multiple servers



A microservices architecture puts each element of functionality into a separate service...



... and scales by distributing these services across servers, replicating as needed.



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

Why Microservices?

- Expectations have changed regarding the delivery of software:
  - Rapid-change and rapid delivery
  - High scalability and reliability
  - Cloud-based

- **Microservice-**based architectures enable the incredible scale and agility needed in the software solutions today

Why Microservices?

Top benefits of adopting microservices

Greater agility

Continuous integration and deployment

Improved scalability

Faster time-to-market

Higher developer productivity

Easier debugging and maintenance

Microservice Attributes

- Microservices aim to be as **decoupled** and as **cohesive** as possible - they own their own domain logic

- Microservices use HTTP request-response with resource **API's** and lightweight messaging

- The messaging infrastructure chosen is typically dumb (message router only, i.e. RabbitMQ) the ***smarts*** live in the end points that are producing and consuming messages; in the “services”

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

Microservice Attributes continued

- Microservices have many **service contracts** to manage; they use simple tools that allow them to define the contract for a service. This becomes part of the automated build before code for the new service is even written.

- Microservices prefer letting each service manage its own database, either different instances of the same database technology, or entirely different database systems - an approach called **Polyglot Persistence**

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

Microservices Characteristics

Decentralized Data Management

Decentralized Governance

Design for Failure

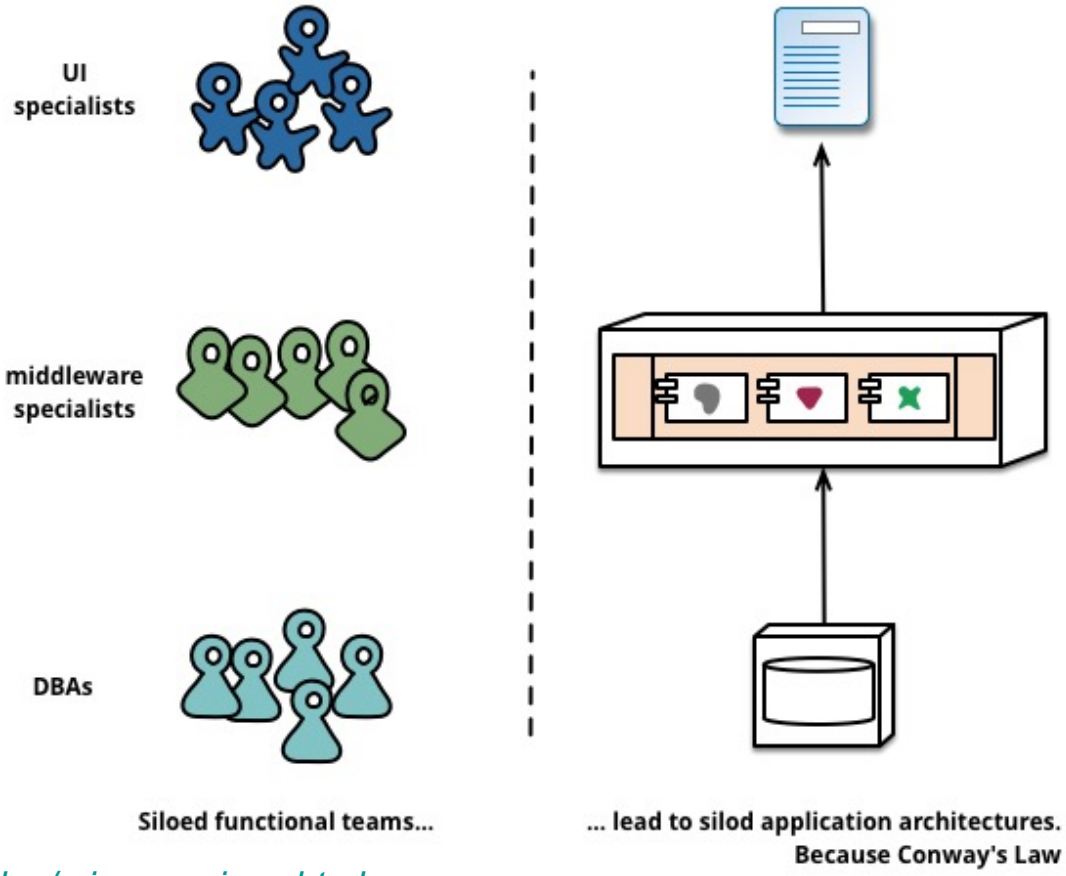
Infrastructure Automation
IT'S AN API!

Evolutionary Design

Conway's Law in action through Microservices

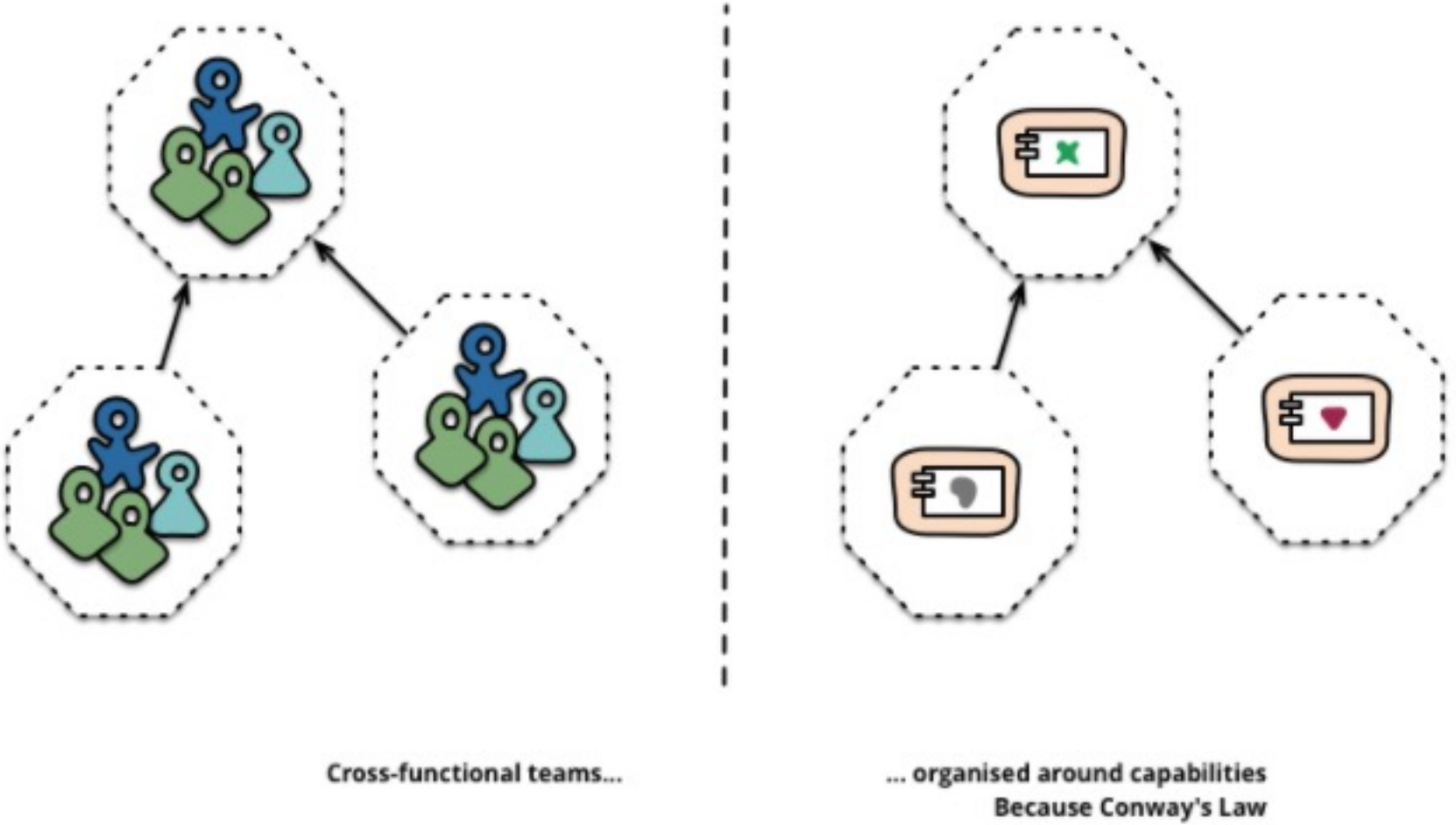
Conway's Law in Action

> *Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure.*
>
> -- *Melvin Conway, 1968*



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

Microservices are Business Services



Cross-functional teams...

... organised around capabilities
Because Conway's Law

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

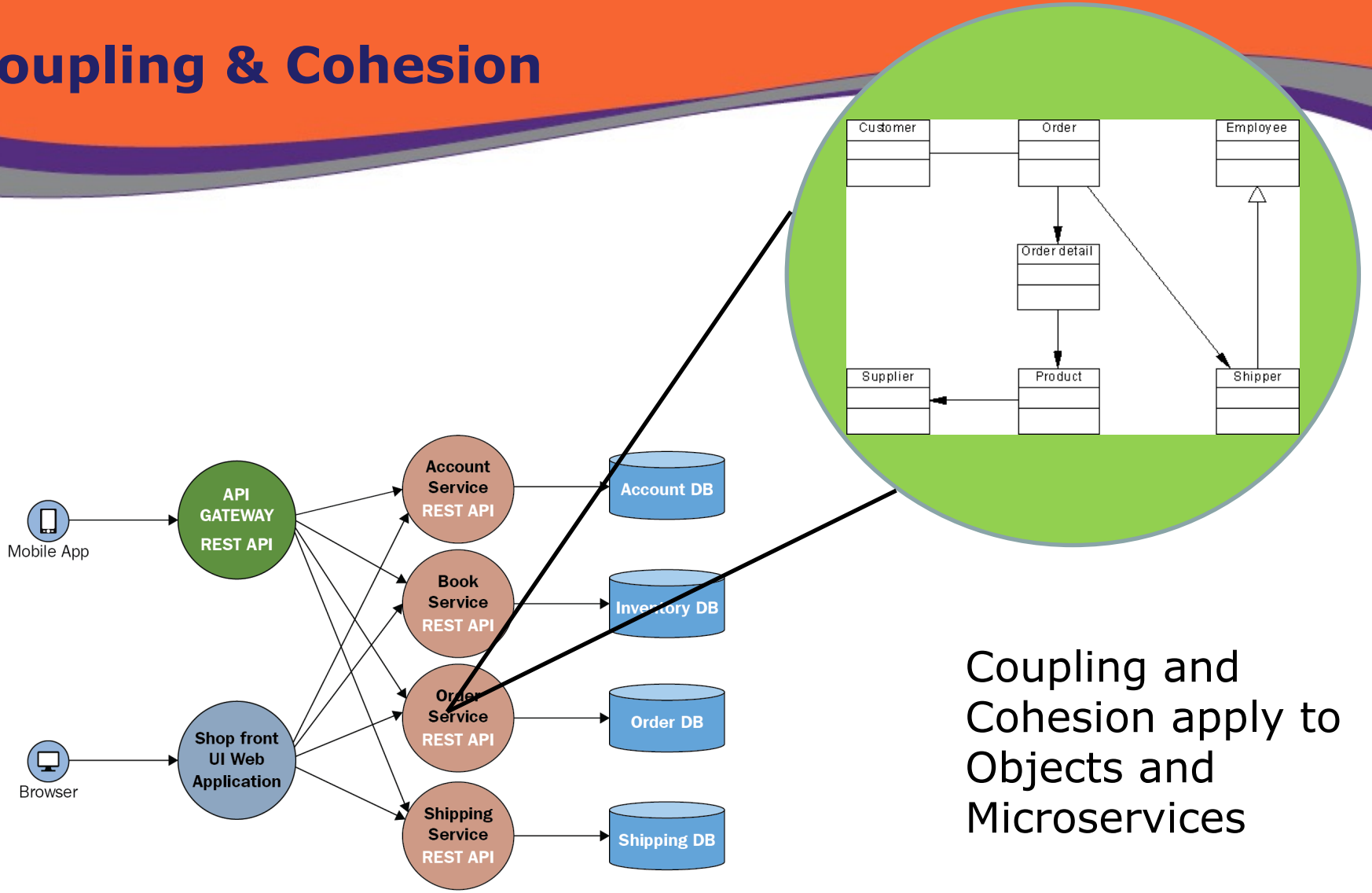
Coupling & Cohesion

Microservices aim to be as **decoupled** and as **cohesive** as possible - they own their own domain logic.

High Cohesion

Low Coupling

Coupling & Cohesion



Coupling and Cohesion apply to Objects and Microservices

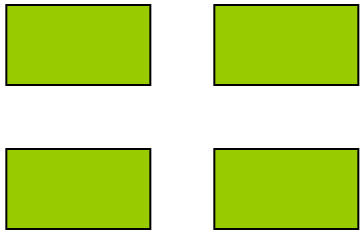
Cohesion

- Definition
  - The degree to which all elements of a component are directed towards a single task.
  - The degree to which all elements directed towards a task are contained in a single component.
  - The degree to which all responsibilities of a single class are related.

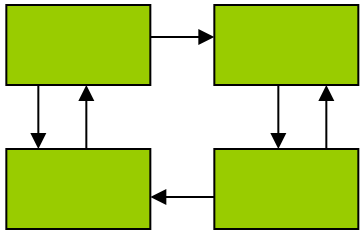
- **High Cohesion**: All elements of a component are directed toward and essential for performing the same task.

Coupling

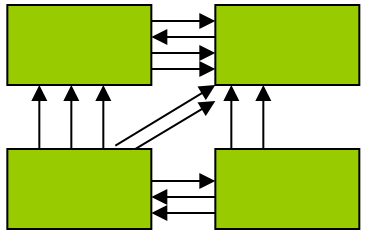
The degree of dependence across components such as the amount of interactions among components



No dependencies



Loosely coupled
some dependencies



Highly coupled
many dependencies

Discussion

What is the effect of cohesion and coupling on maintenance?

Consequences of Coupling

- High coupling
  - Components are difficult to understand in isolation
  - Changes in component ripple to others
  - Components are difficult to reuse
    - Need to include all coupled components
    - Difficult to understand
- Low coupling
  - May incur performance cost
  - Generally faster to build systems with low coupling

Let's take an example....

Clemson Online Bank: Epics

1. As a bank user, I want to login to my account.
2. As a protentional bank customer, I want to signup for an account.
3. As a bank user, I want to edit my account information.
4. As a bank user, I want to transfer money to another Clemson Bank account
5. As a bank user I want to transfer money within my own account.
6. As a bank user, I want to transfer money to a non-Clemson account.
7. As a bank user, I want to be notified if I have insufficient funds in my account.
8. As a user, I want to deposit a check online
9. As a bank user, I want to set up travel notifications
10. As a bank user, I want to receive notifications of suspicious transaction activity.
11. As a bank user, I want to see dashboards of my spending.
12. As a bank user, I want to replace a credit card that I lost.
13. As a bank user, I want to lock or unlock credit cards.
14. As a bank user, I want to see transaction history.

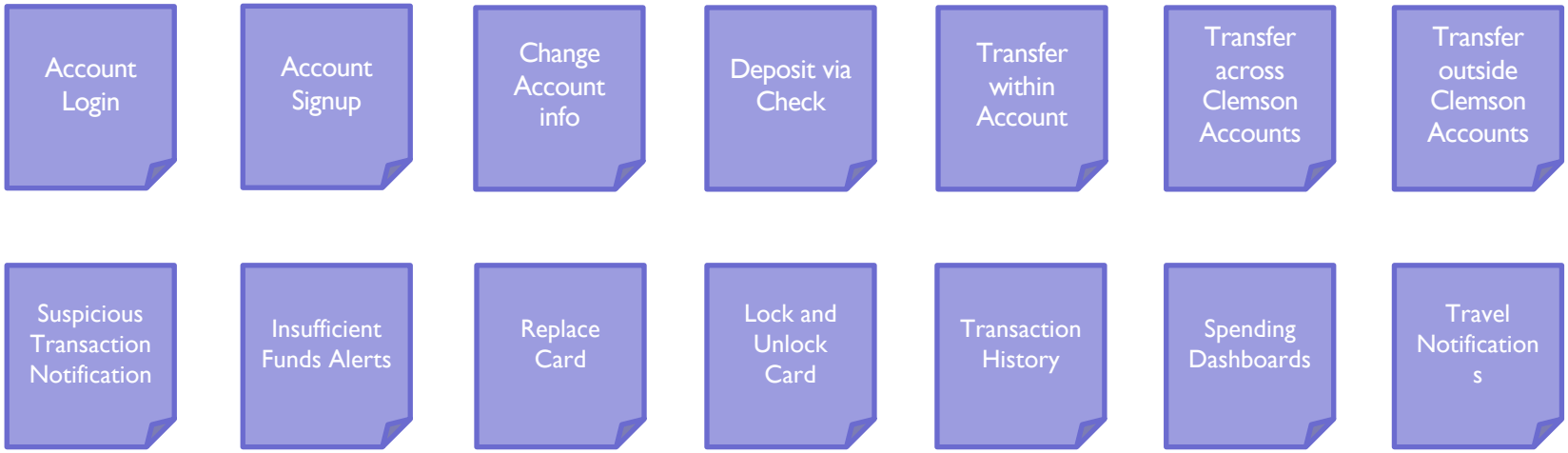
How would you break up the Clemson Bank into Services?

With your teams and brainstorm a list of “Services” that would make up the component/microservices of the Bank to support these Epics. Have a list to share in 10 minutes.

>>>>>

Let's take an example....

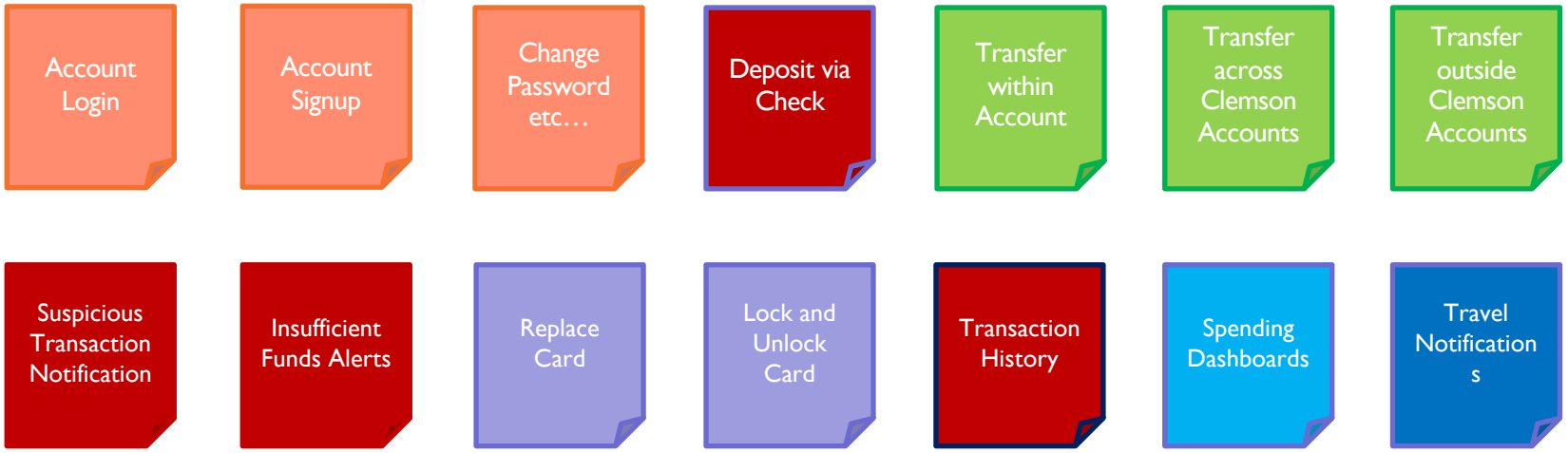
Clemson Online Bank
Epics



Grouping of similar requirements..

An online banking system……

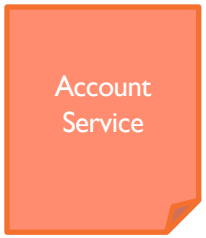
Clemson Bank

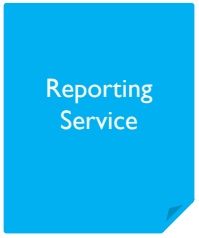


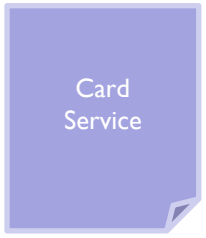
Services

An online banking system……

Clemson Bank







Upcoming

- **Quiz 2 on Thursday!** Same drill as Quiz 1. Lessons 5-7 (mainly user stories).
- **Sprint 0 Review Tuesday 2/13 !**
- **Microservices Assignment - due end of today!**