

ThoughtWorks®

Hola!

MICRO SERVICES ARCHITECTURE

*South Sound Developers User Group
Olympia, Washington
March 13, 2015*

MY FIRST CONTROLLER CLASS

[illegible]

MY FIRST CONTROLLER CLASS

Authentication

Authentication

Process Shopping Cart

Data Access Logic

Process Shipping Information

Data Access Logic

Presentation Logic

Data Access Logic

Process Payments

[illegible]

*So what's wrong with **that**?*

MY FIRST CONTROLLER CLASS



“Good design scales up well.”

MICRO SERVICES ARCHITECTURE

- ☐ Key Traits of MSA
- ☐ Design Principles
- ☐ Walk-through
- ☐ Implementation Approach

MICRO SERVICES ARCHITECTURE

- ☐ Technical Details
 - ☐ Messaging
 - ☐ Monitoring
 - ☐ Testing
 - ☐ Deployment

MICRO SERVICES ARCHITECTURE

- ☐ Pros and Cons
- ☐ References
- ☐ Questions (maybe some answers)

Ready?

KEY TRAITS

☐ Independent

- ☐ Build
- ☐ Deployment
- ☐ Data Management
- ☐ Replaceable
- ☐ Business Domain

□ SOLID Design for Distributed Systems

- Single Responsibility
- Open / Closed
- Liskov Substitution
- Interface Segregation
- Dependency Inversion

□ Unix Philosophy

KEY TRAITS

☐ Robust

- ☐ Loosely Coupled
- ☐ Decentralized
- ☐ Stateless
- ☐ Asynchronous
- ☐ Self Monitoring

KEY TRAITS

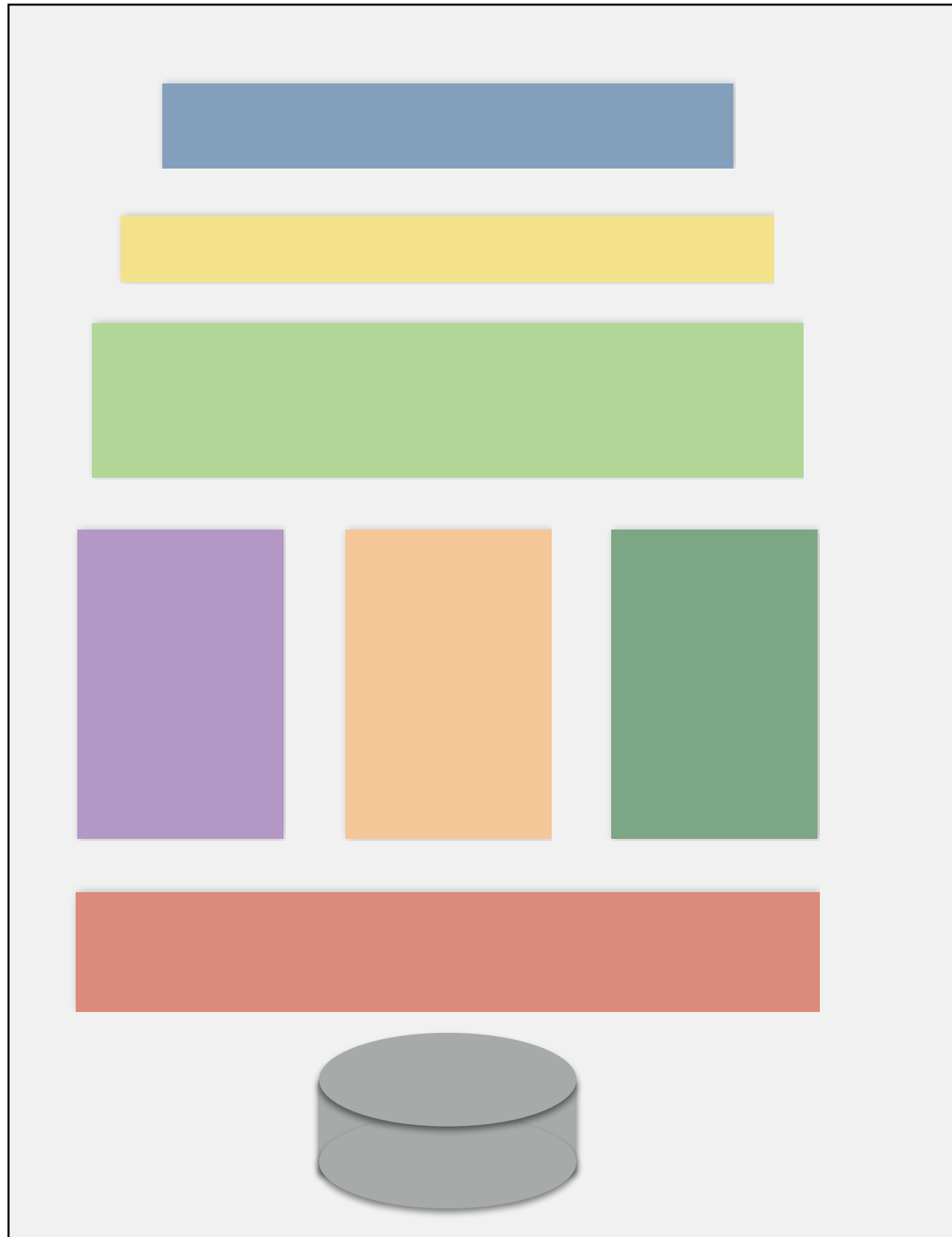
☐ Agile

- ☐ Fast cycle time
- ☐ Focused on value
- ☐ Responsive to change
- ☐ Emergent design
- ☐ Short feedback loop on value
- ☐ Easier to experiment

Conway's Law:

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

THE MONOLITH



Better?

- ☐ Separation of concerns
- ☐ Testable
- ☐ Stable

But...

- ☐ Changes require full re-deploy
- ☐ Still somewhat coupled
- ☐ Large codebase - harder to keep organized

SERVICES - THE SAGA BEGINS



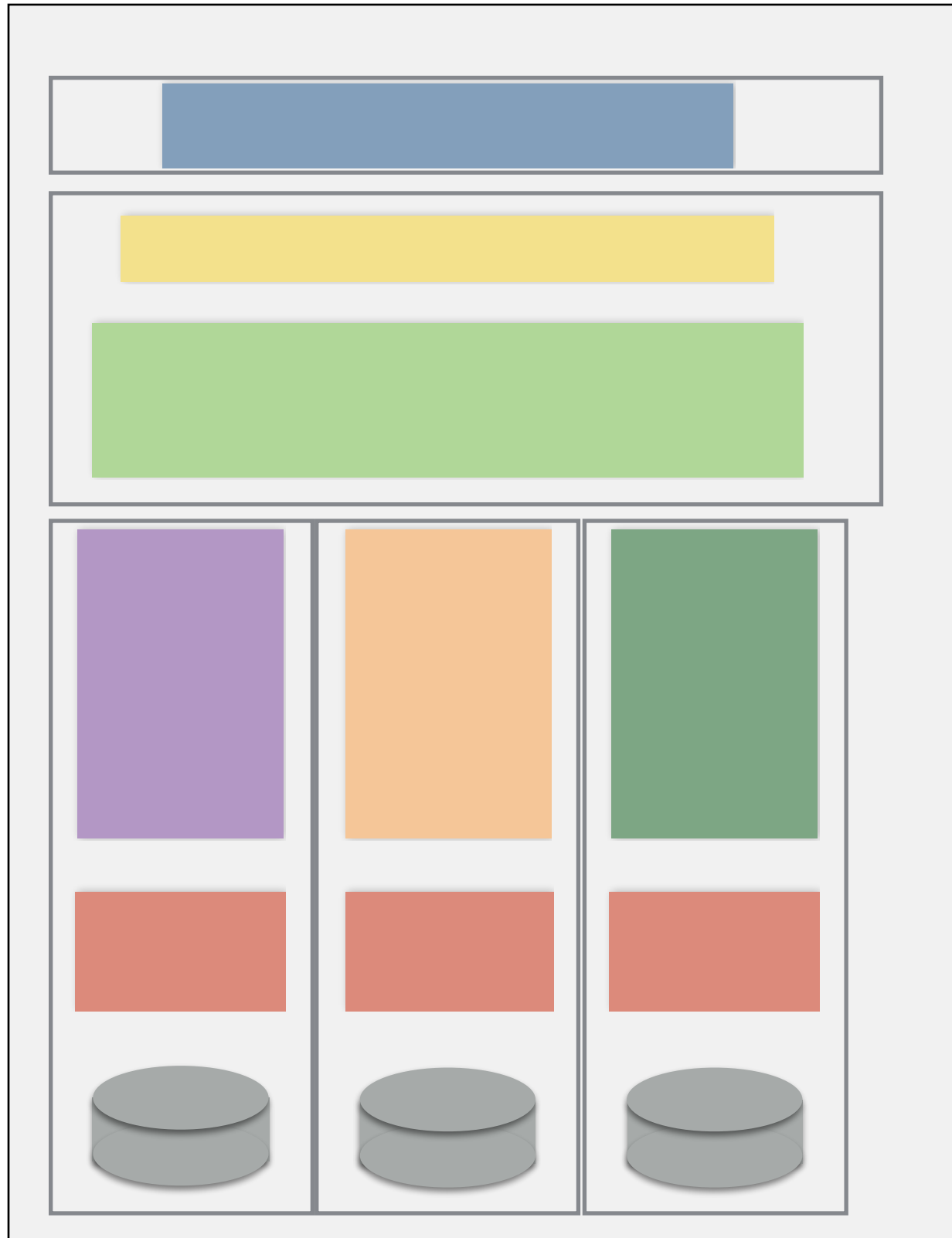
Better?

- ☐ Independently deployable
- ☐ Testable
- ☐ Stable

But...

- ☐ Contracts and Versioning
- ☐ Centralized Database - Sql hell
- ☐ Transactions and Concurrency
- ☐ Latency

SERVICES - THE SAGA CONTINUES



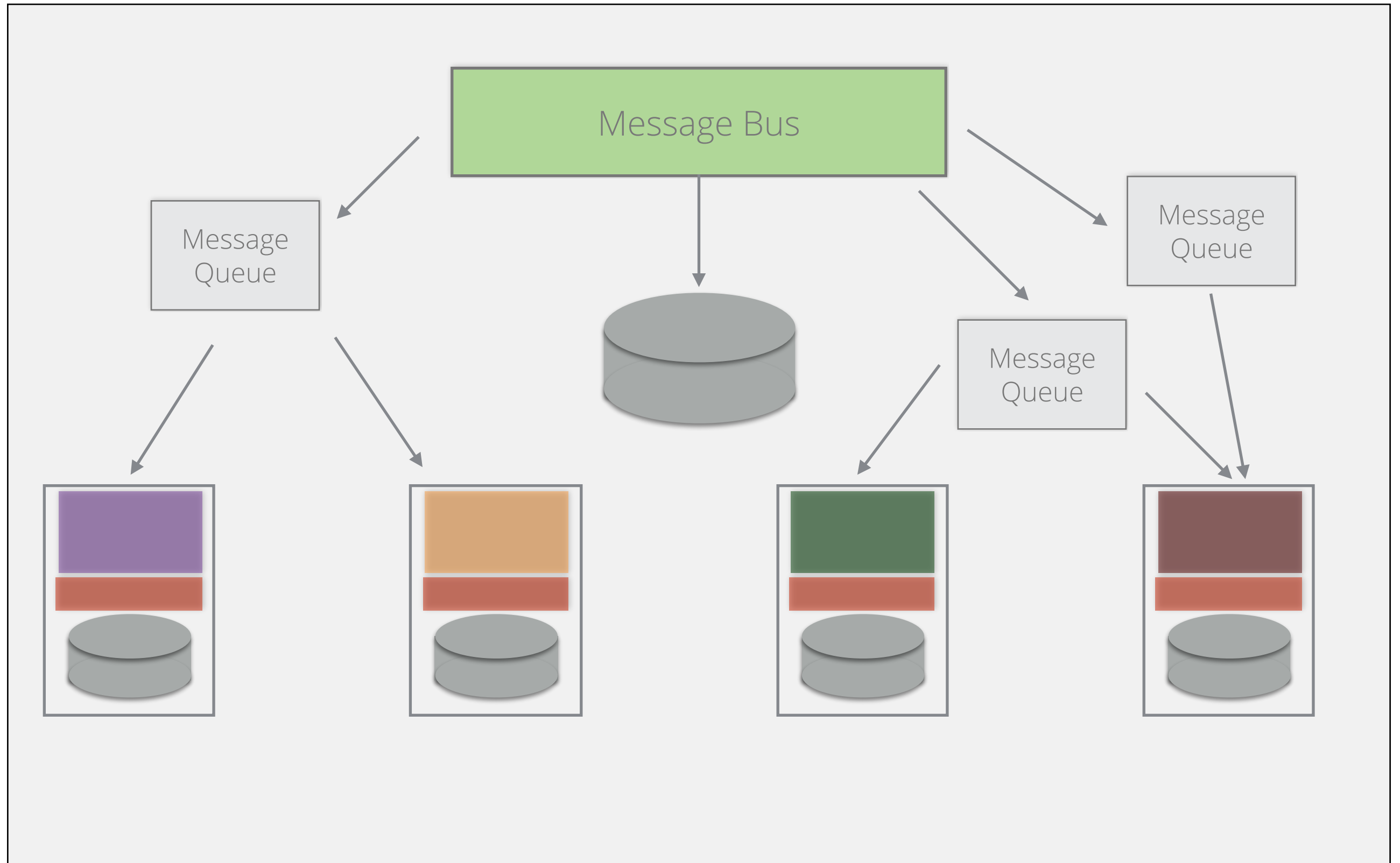
Better?

- ☐ Independently deployable
- ☐ Testable
- ☐ Stable

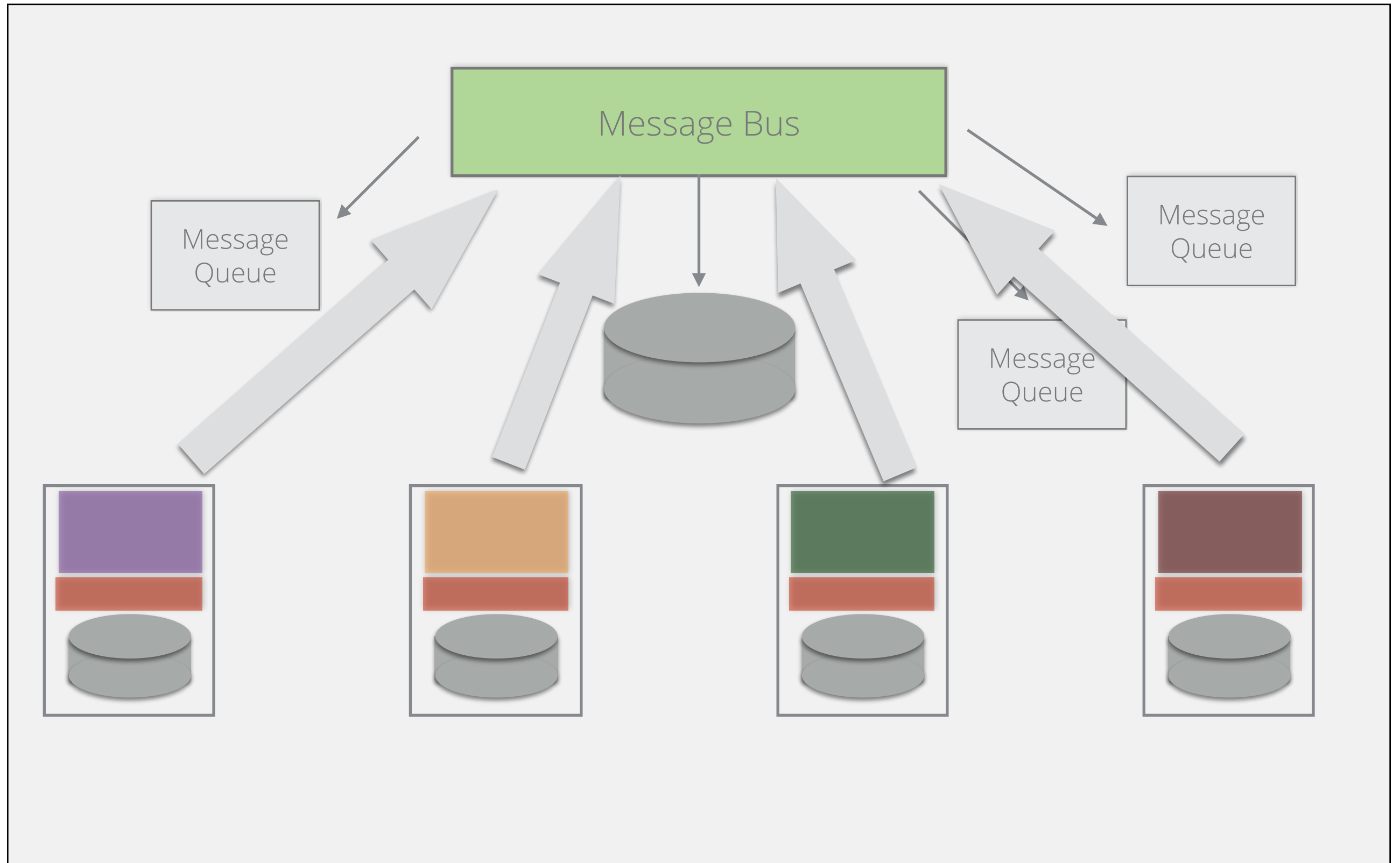
But...

- ☐ Contracts and Versioning
- ☐ Orchestration
- ☐ DB Synchronization
- ☐ Services still somewhat coupled

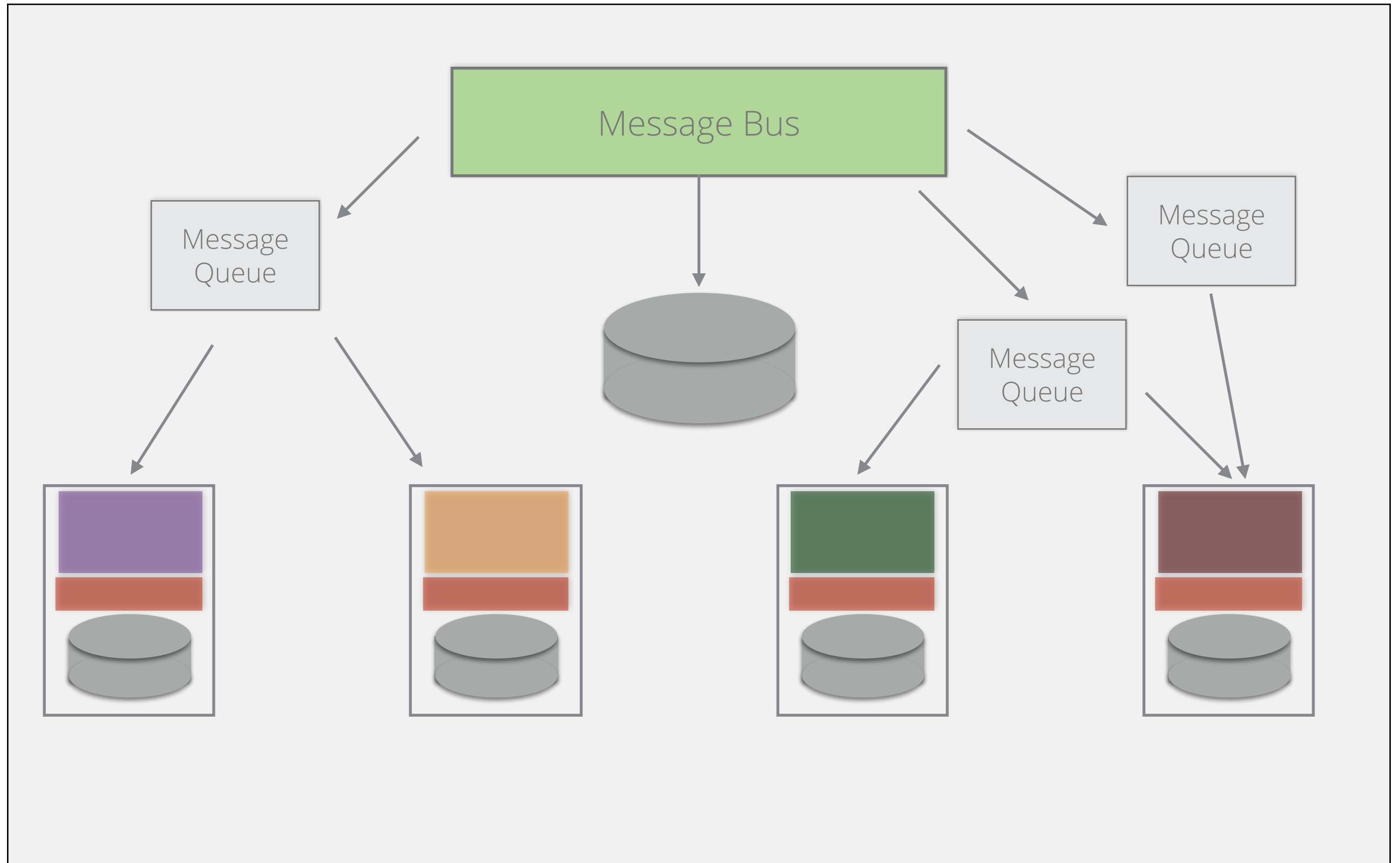
A STEP FURTHER...



A STEP FURTHER...



A STEP FURTHER...



- Report Everything
 - You never know what might be interesting
- Expire messages
- Generous Contracts are generous
- Queues and Listeners
- Async
- Fault tolerant

MONITORING

- ❑ Decentralized and async nature means monitoring is important
- ❑ Monitor thyself
- ❑ Fail loudly
- ❑ Performance metrics
- ❑ Business metrics
- ❑ Session Tracking
- ❑ Security Logging

TESTING

- ❑ Small services make unit testing nice
- ❑ Async nature and messages make end to end testing less relevant
- ❑ Fault tolerance and intense monitoring boosts confidence in releasing
- ❑ Regression testing?
- ❑ A/B Testing and experiments?

DEPLOYMENT

- ☐ Continuous Integration / Continuous Delivery
- ☐ Frequent releases
- ☐ Low Ceremony
- ☐ No “flipping switches”

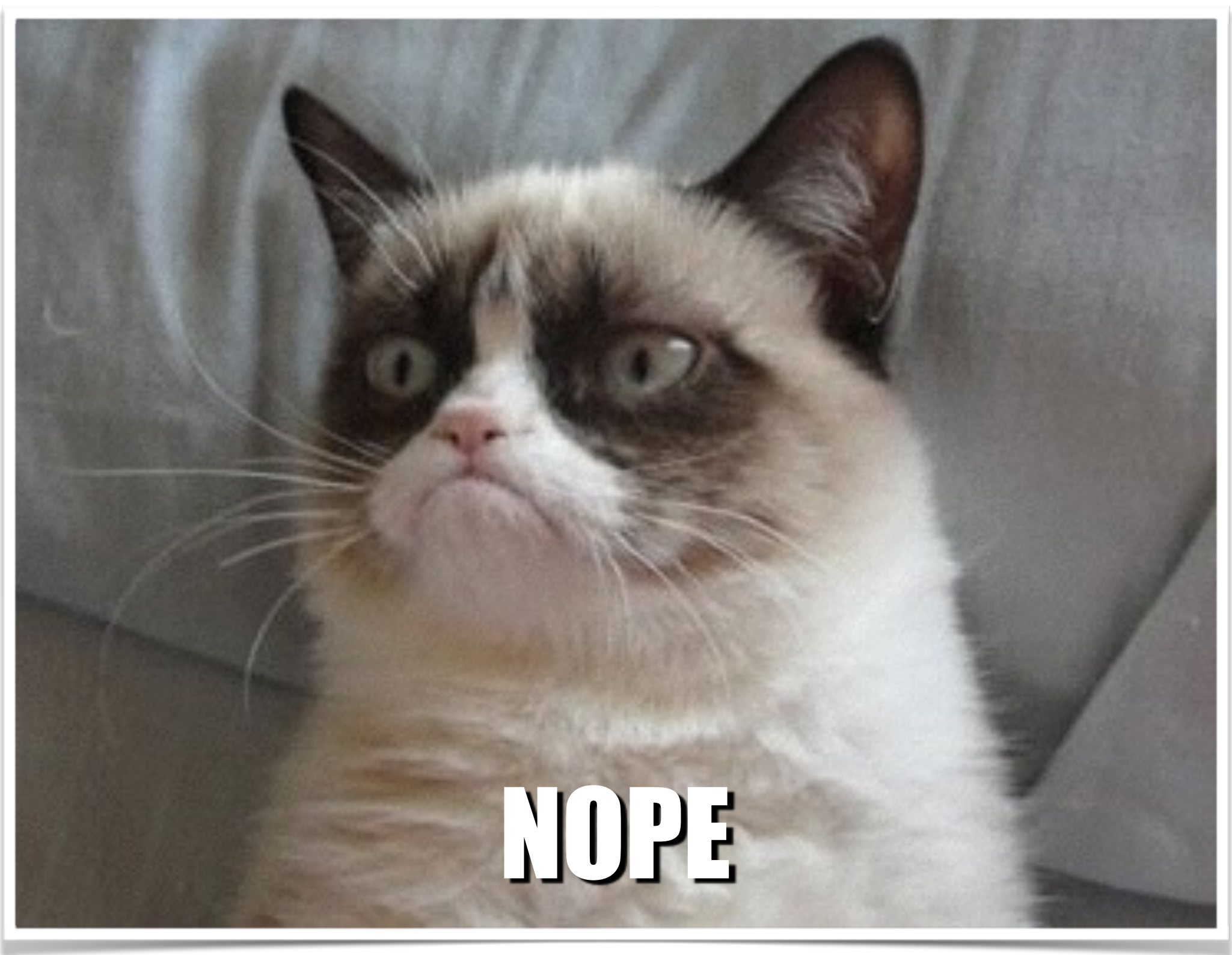
PROS

- Robust
- Simple
- Scalable
- Testable
- Composition of systems
- Easy to change
- Easy to replace
- Easy to deploy

PROS

- ❑ Language agnostic
- ❑ Platform agnostic
- ❑ Framework agnostic
- ❑ You're not locked in to legacy code!

“I get it, Micro Services is a Silver Bullet!”



CONS

- ❑ Requires robust monitoring
- ❑ More services to keep track of
- ❑ Not an intuitive mental model...yet
- ❑ Finding domain boundaries may take trial and error

REFERENCES

- Martin Fowler's article:

 - <http://martinfowler.com/articles/microservices.html>

- James Hughes:

 - <http://yobriefca.se/blog/2013/04/29/micro-service-architecture/>

- Fred George:

 - OreDev video: <https://vimeo.com/79866979>

- also, the google knows

THANK YOU

Questions?

*Michael Ibarra
mibarra@thoughtworks.com
@bm2yogi*

ThoughtWorks®