

ARCHITECTING FOR CONTINUOUS DELIVERY

Ken Mugrage – ThoughtWorks Technology Advocate

TEAM STRUCTURE

CONWAY'S LAW

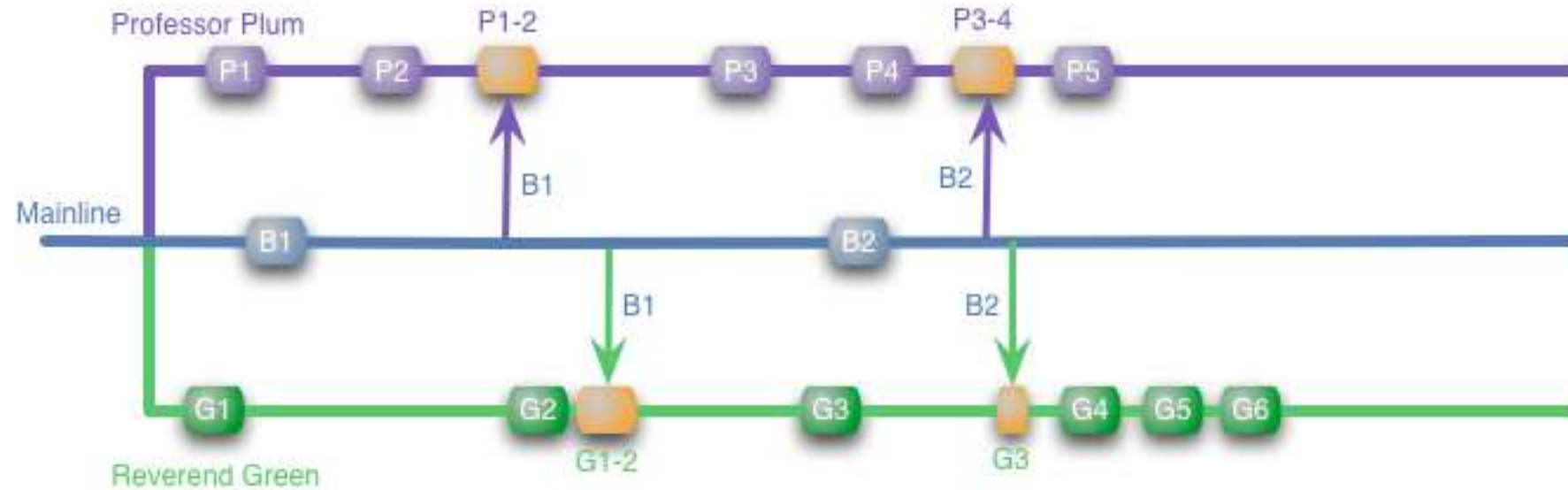
Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure.

YOU BUILD IT, YOU RUN IT

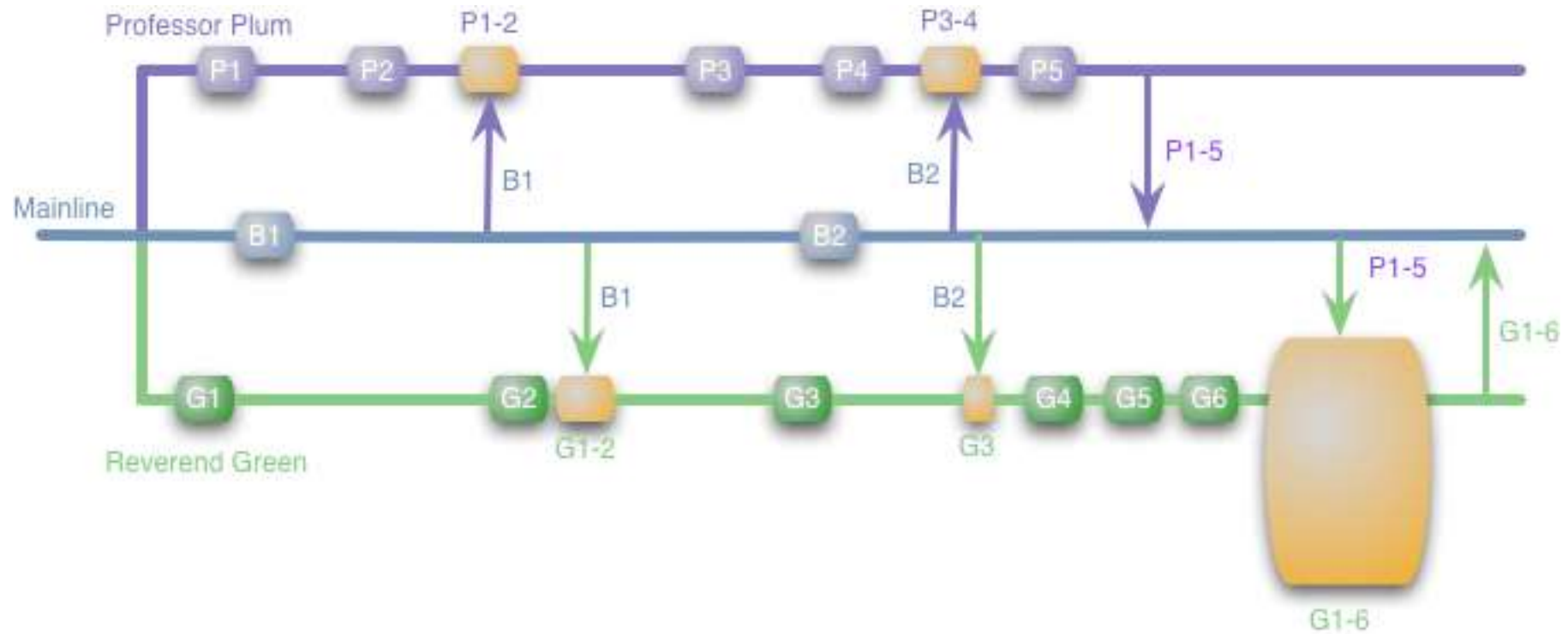
“...It also brings them into day-to-day contact with the customer. This customer feedback loop is essential for improving the quality of the service.”

CODE MANAGEMENT

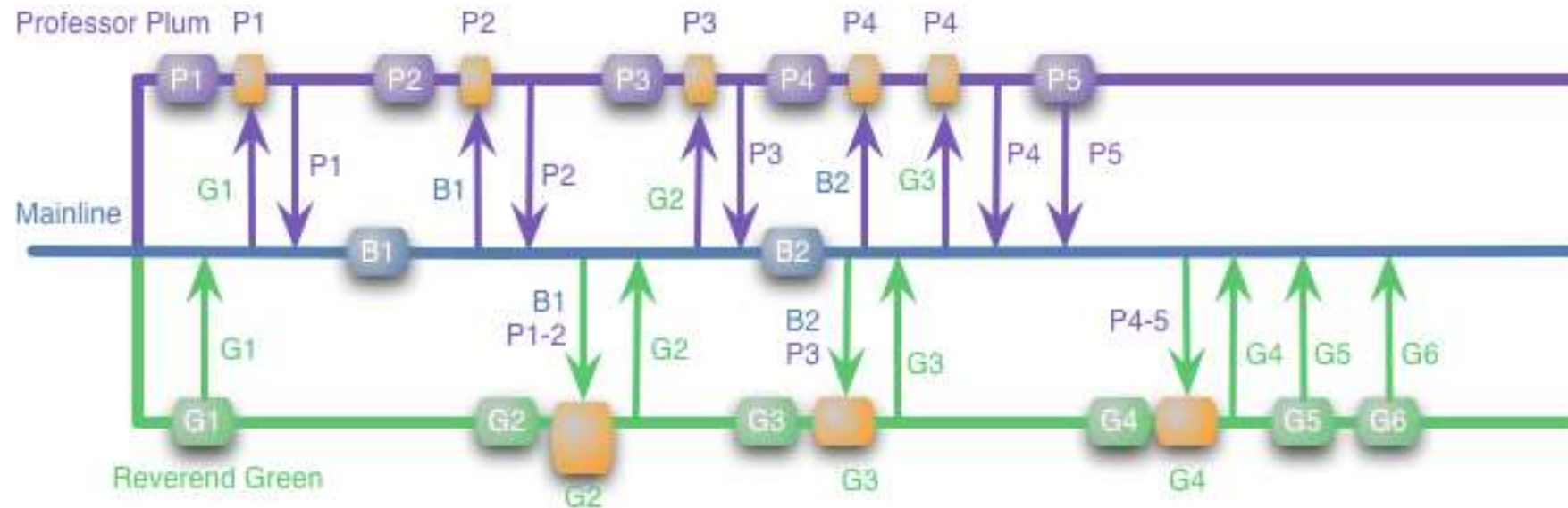
TRUNK BASED DEVELOPMENT



TRUNK BASED DEVELOPMENT



TRUNK BASED DEVELOPMENT



The purpose of a Continuous Delivery pipeline is to kill release candidates

FEATURE TOGGLES

```
function reticulateSplines() {  
    // current implementation lives here  
}
```

FEATURE TOGGLES

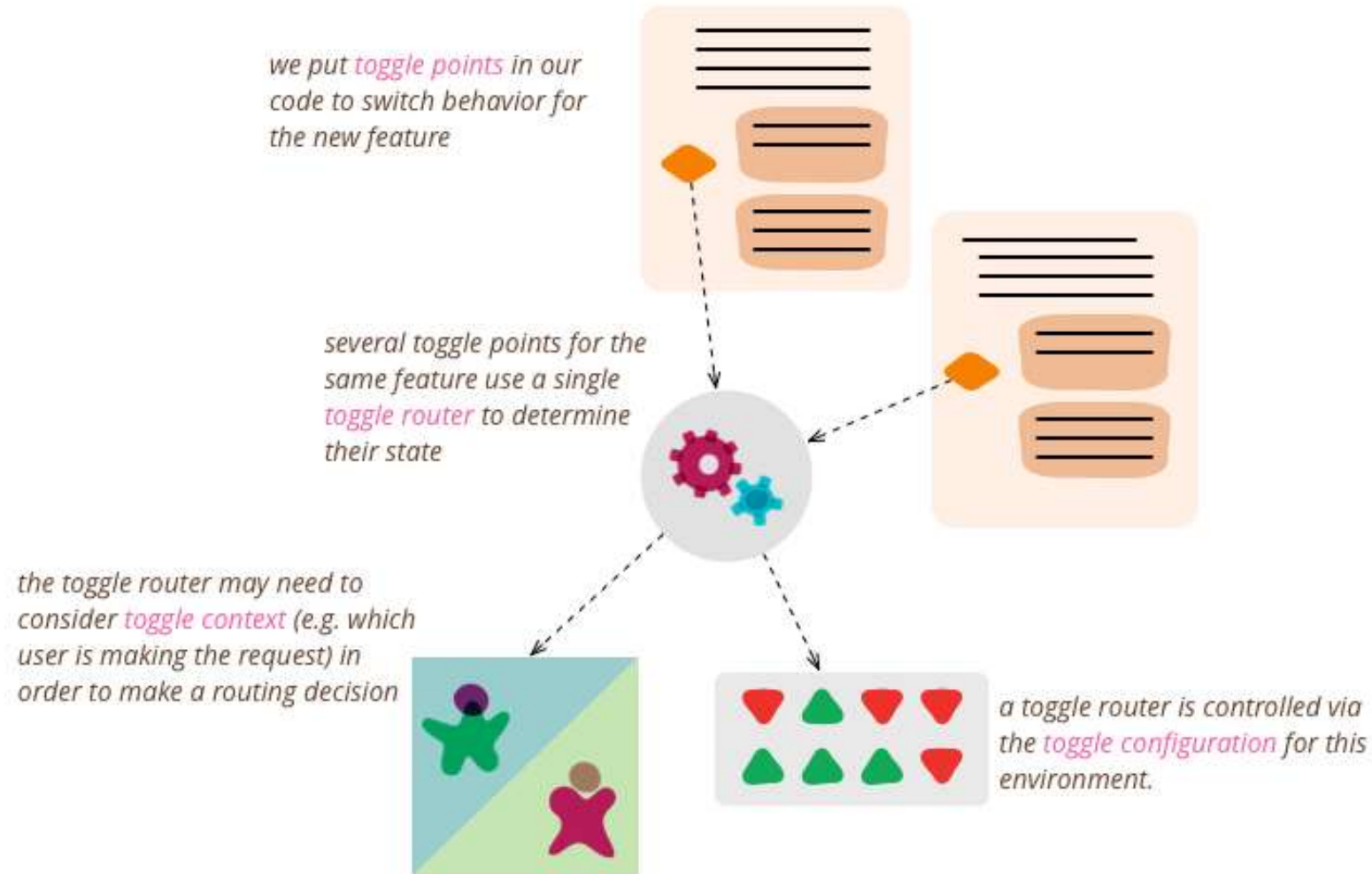
```
function reticulateSplines(){
  var useNewAlgorithm = false;
  // useNewAlgorithm = true; // UNCOMMENT IF YOU ARE WORKING ON THE NEW SR ALGORITHM

  if( useNewAlgorithm ){
    return enhancedSplineReticulation();
  } else {
    return oldFashionedSplineReticulation();
  }
}

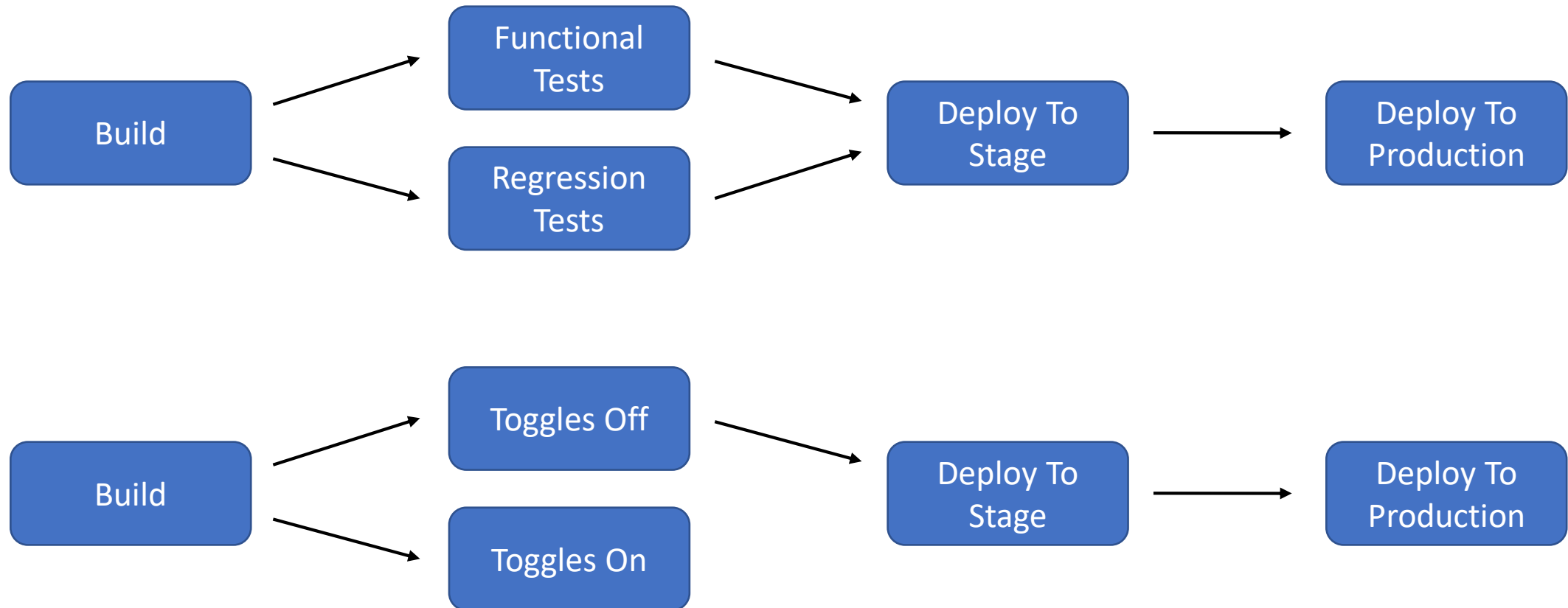
function oldFashionedSplineReticulation(){
  // current implementation lives here
}

function enhancedSplineReticulation(){
  // TODO: implement better SR algorithm
}
```

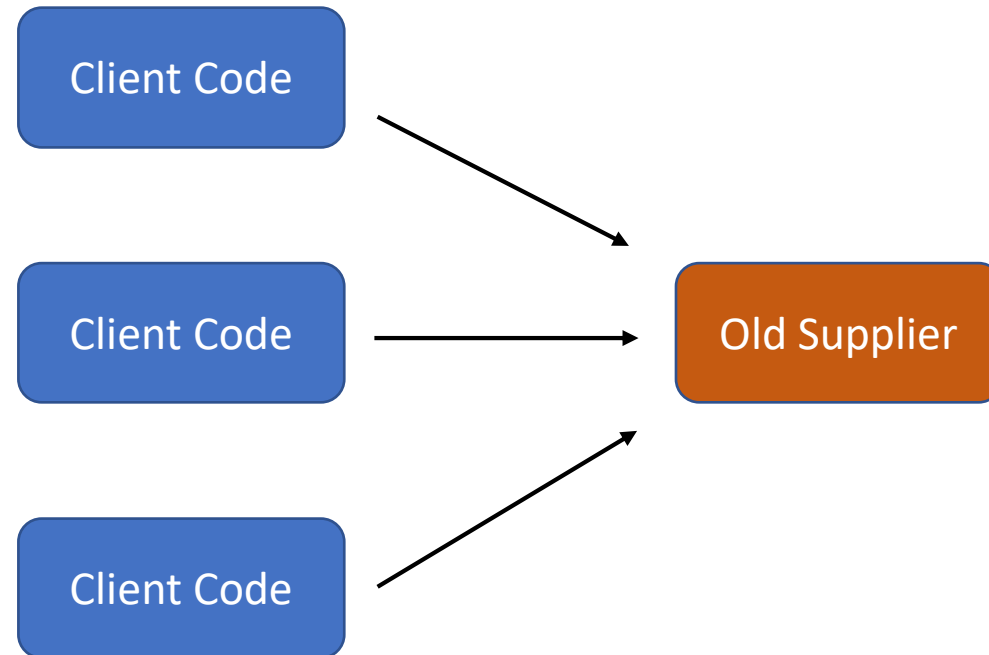
FEATURE TOGGLES



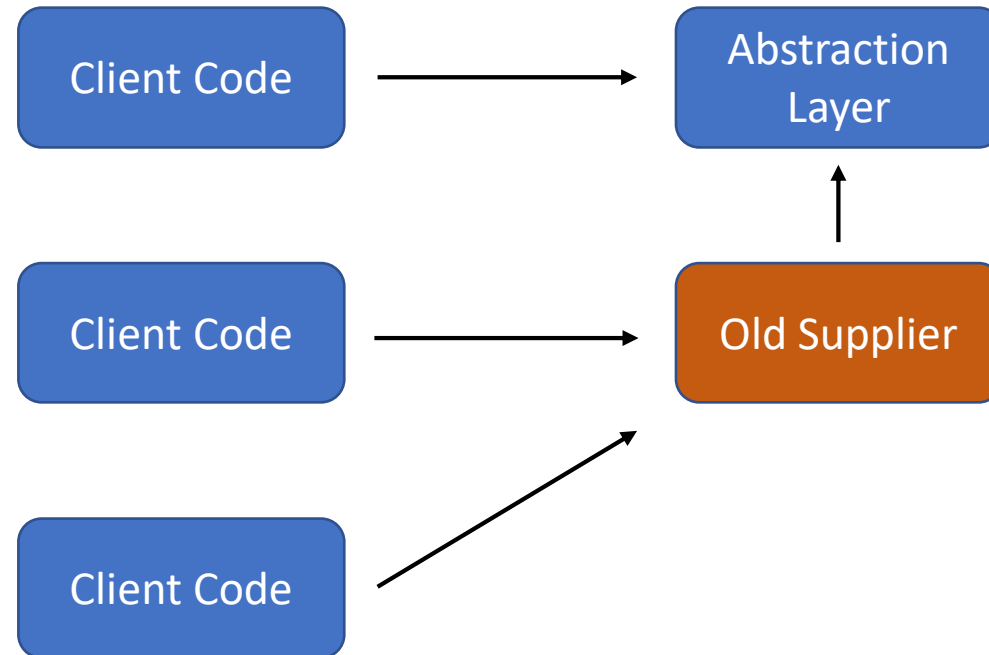
THE PIPELINE



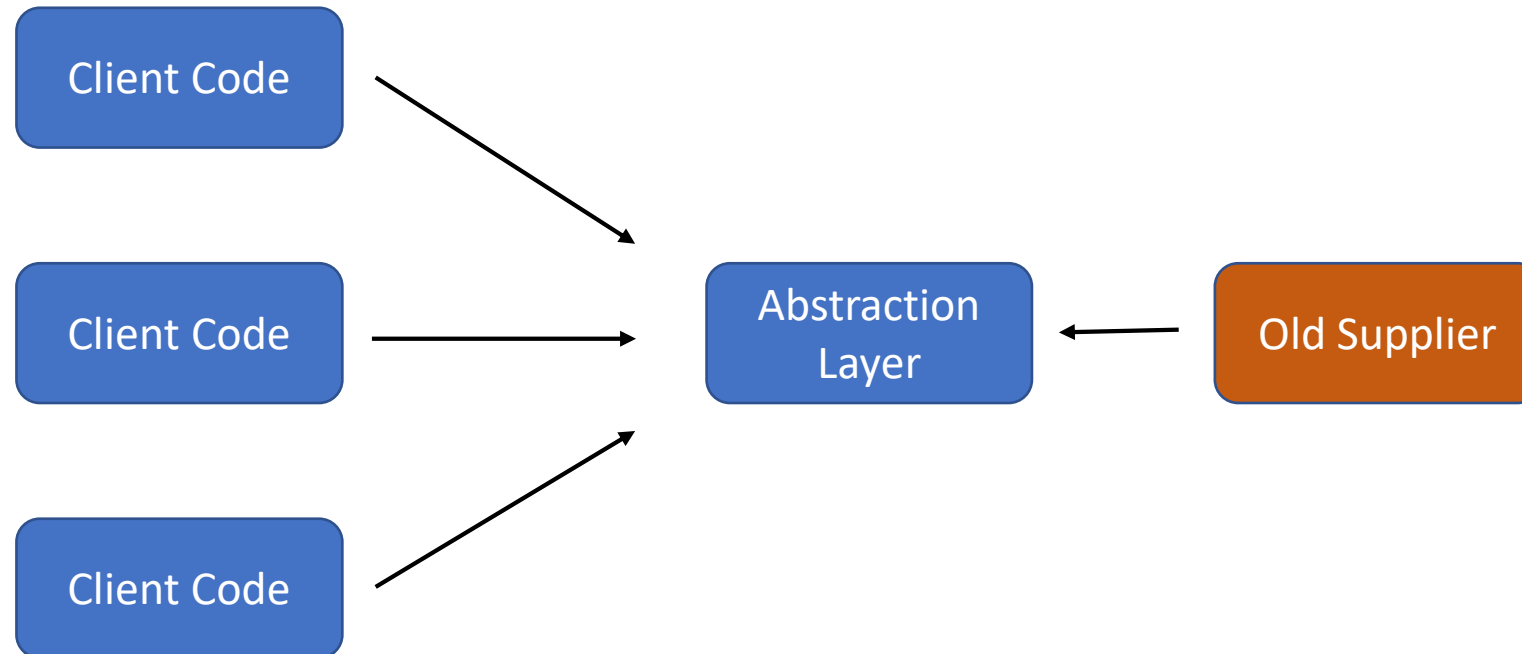
BRANCH BY ABSTRACTION



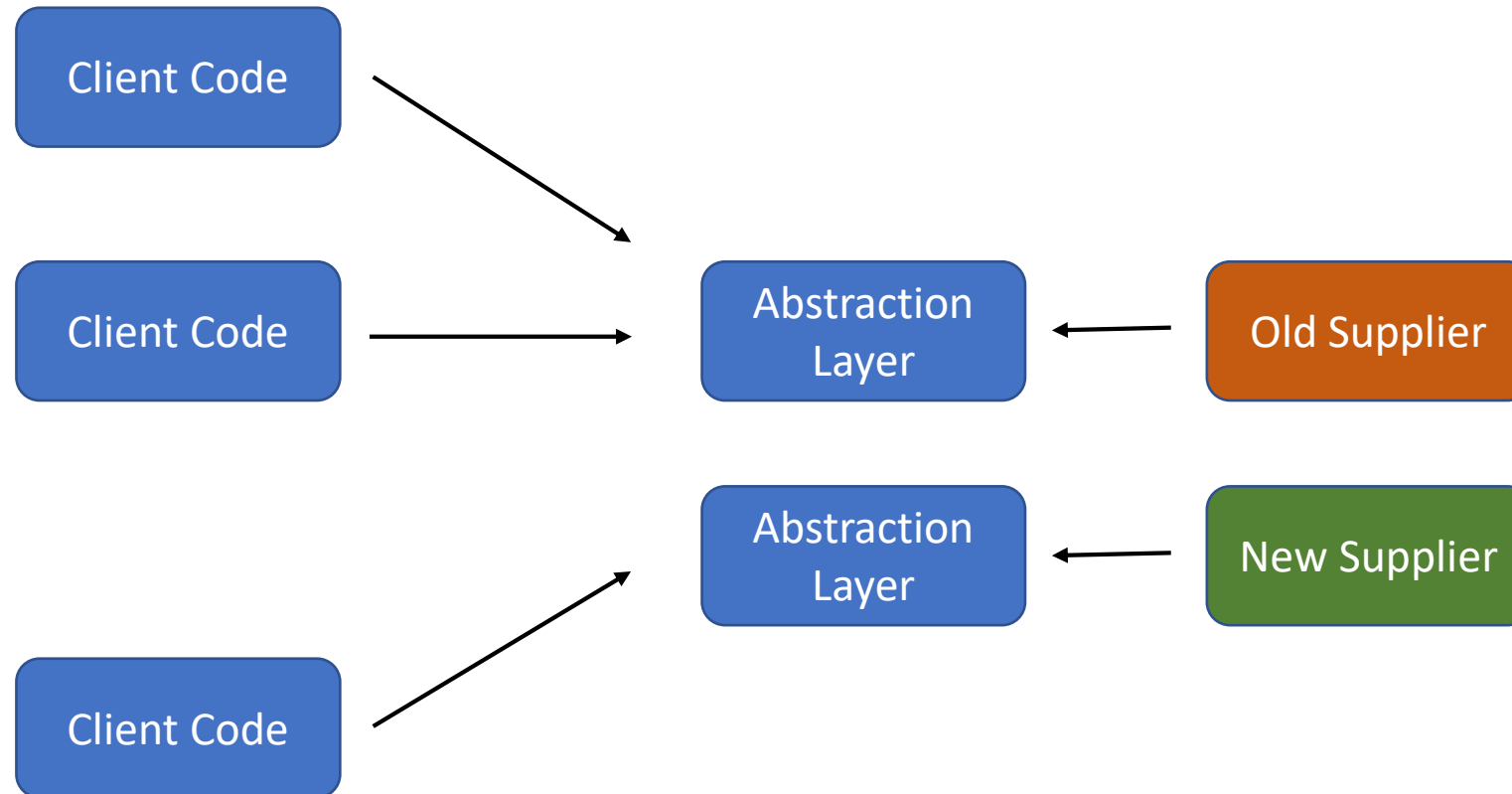
BRANCH BY ABSTRACTION



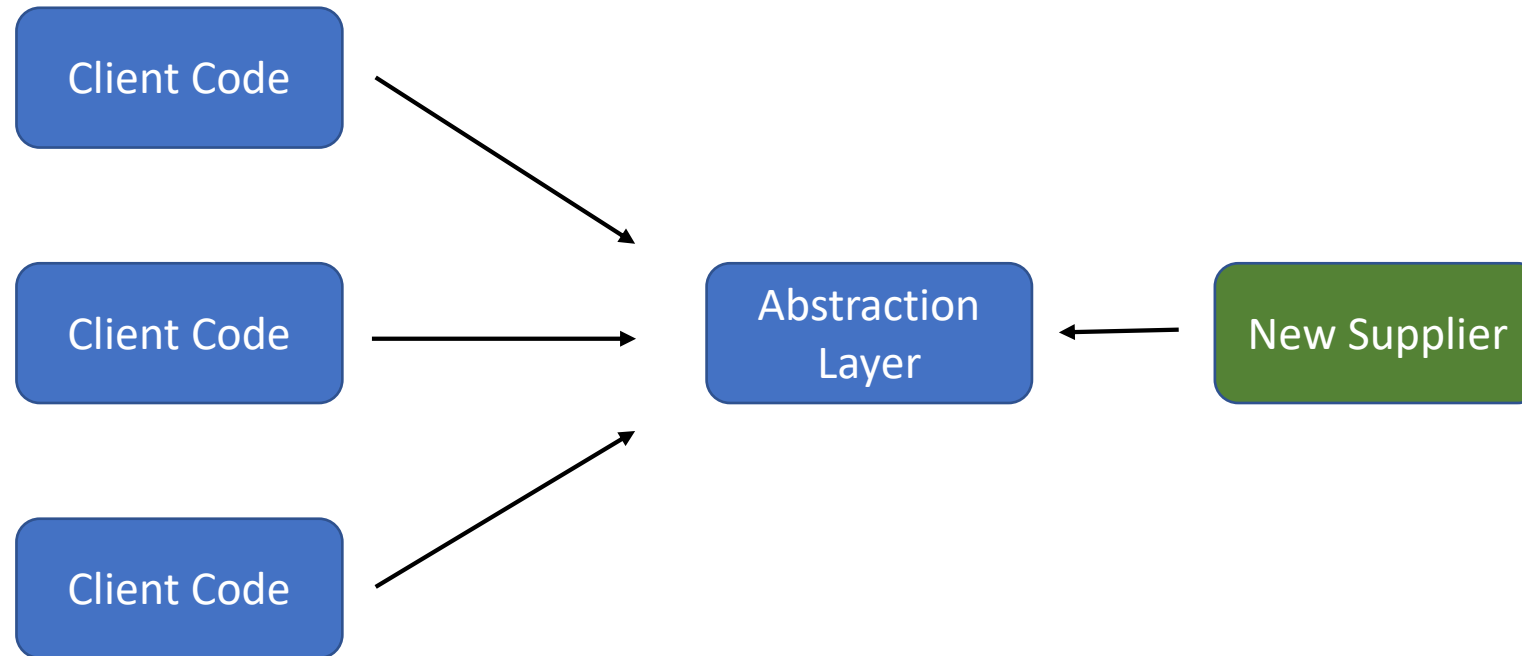
BRANCH BY ABSTRACTION



BRANCH BY ABSTRACTION

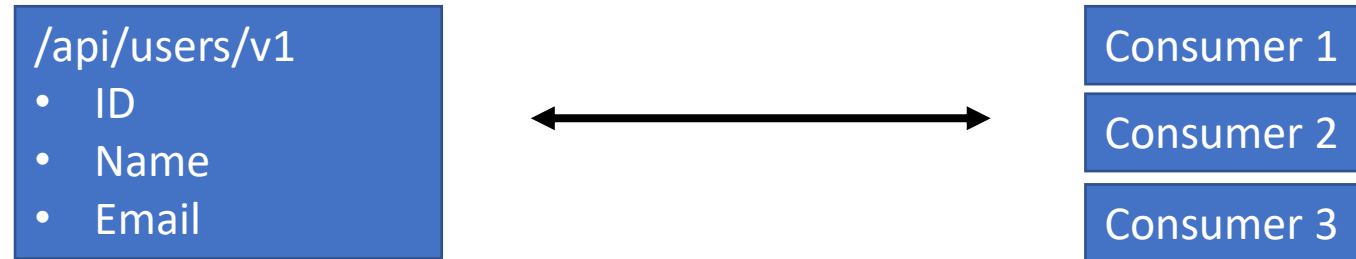


BRANCH BY ABSTRACTION

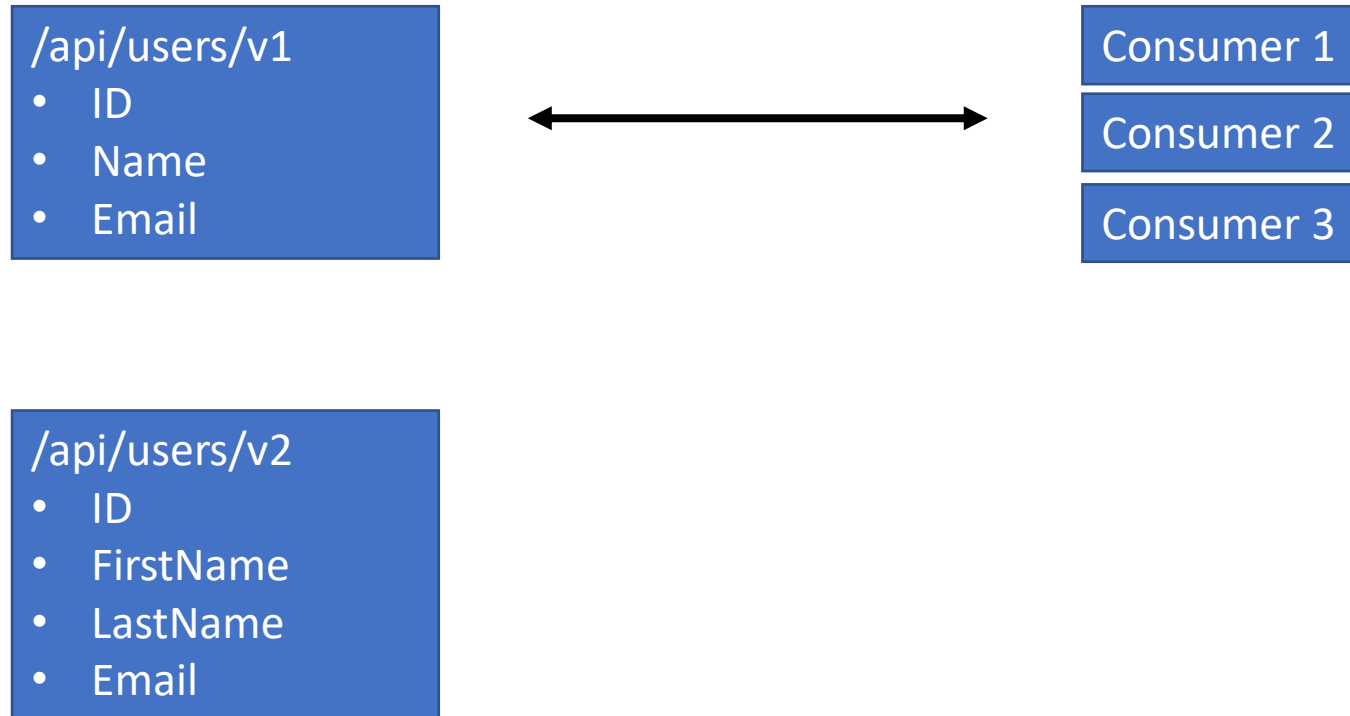


ARCHITECTURE

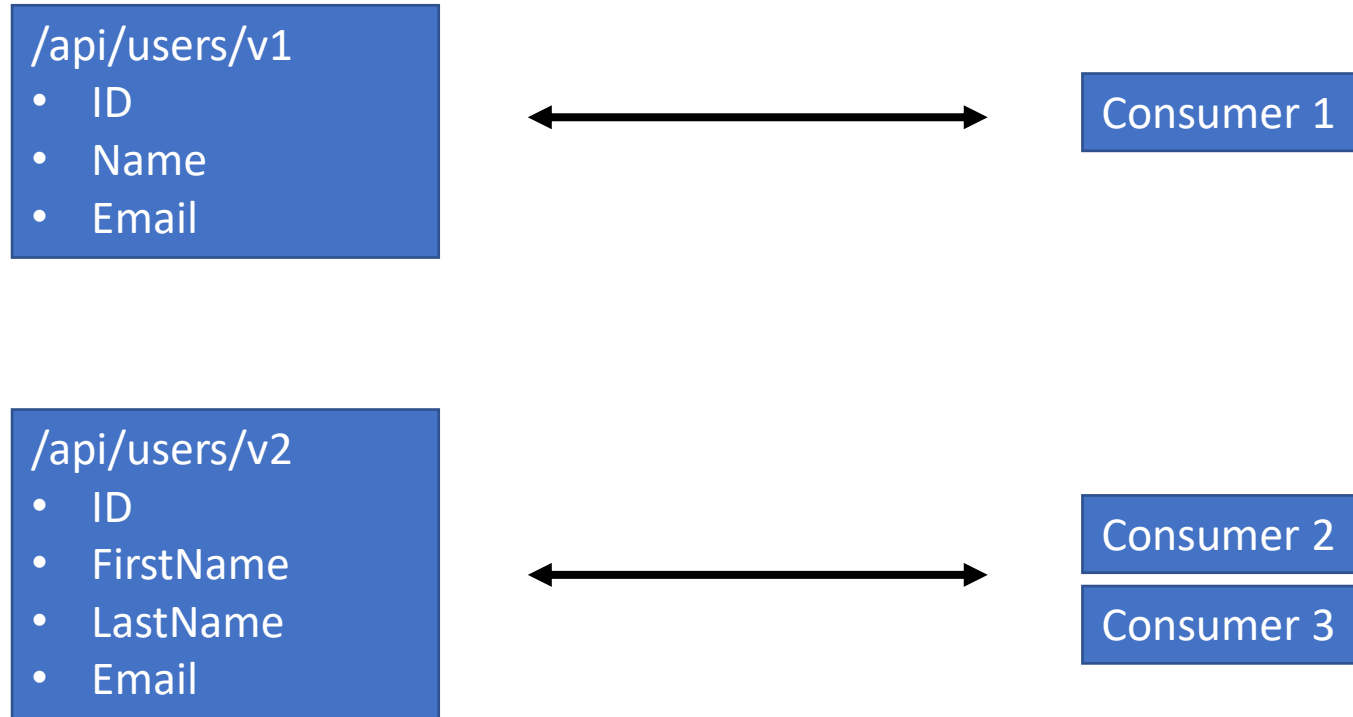
WELL VERSIONED APIS



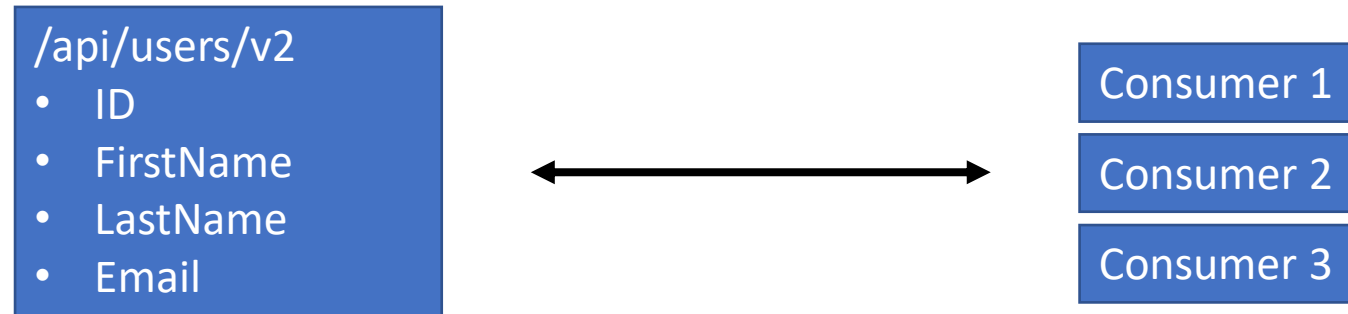
WELL VERSIONED APIS



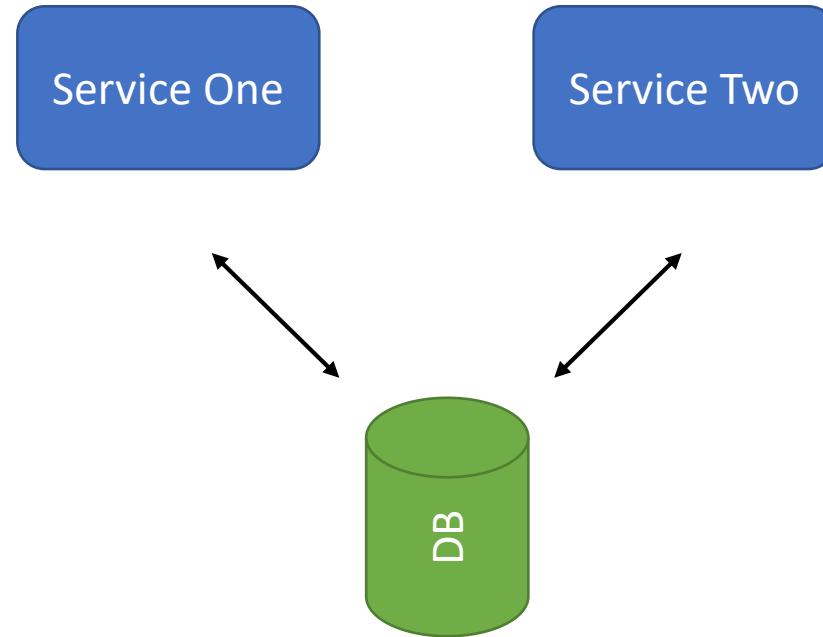
WELL VERSIONED APIS



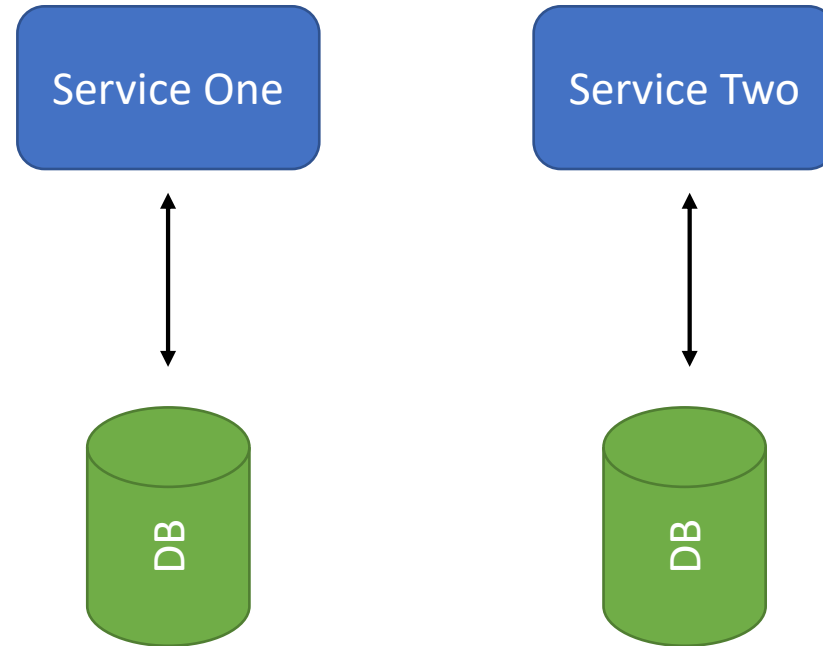
WELL VERSIONED APIS



NO SHARED DATA STORE

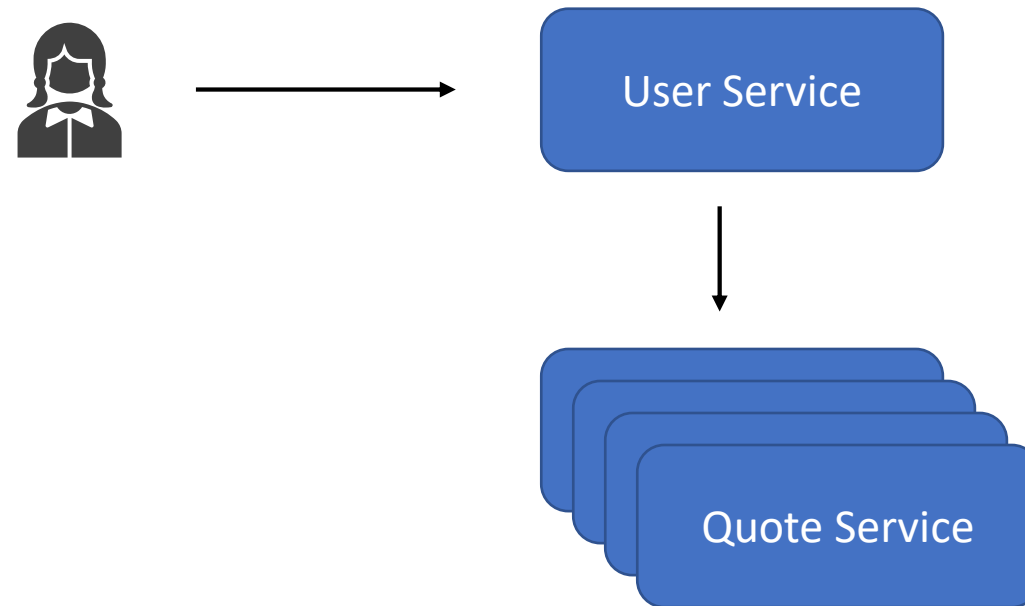


NO SHARED DATA STORE

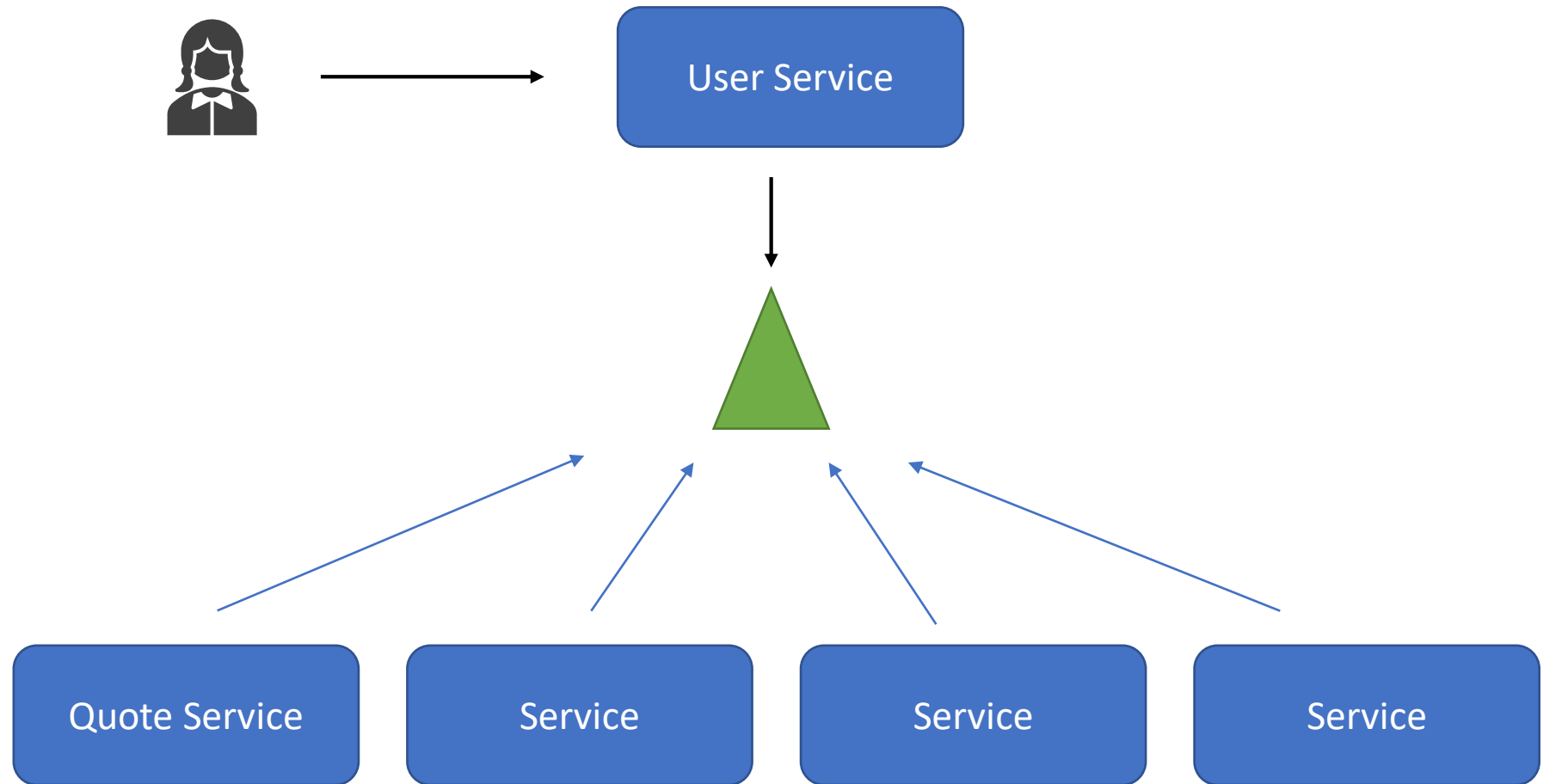


EVENT DRIVEN APPLICATIONS

WITHOUT EVENTS



EVENT NOTIFICATION



EVENT SOURCING

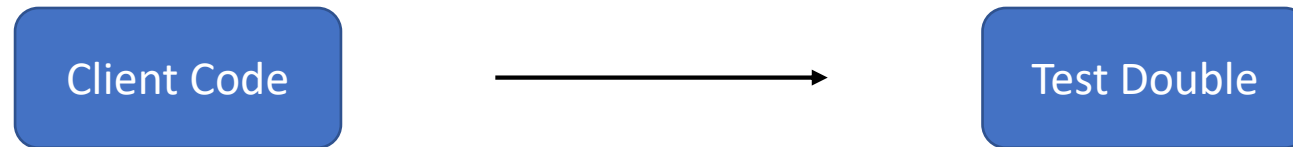
- You don't write to the data store, you create an event which writes to the store
- The test: You could completely blow away the store and recreate it from the event stream
- You use this model every day (I hope)

TESTING DISTRIBUTED APPS

- We're using external services
- We can't reliably test them
- Flaky tests are worse than no tests

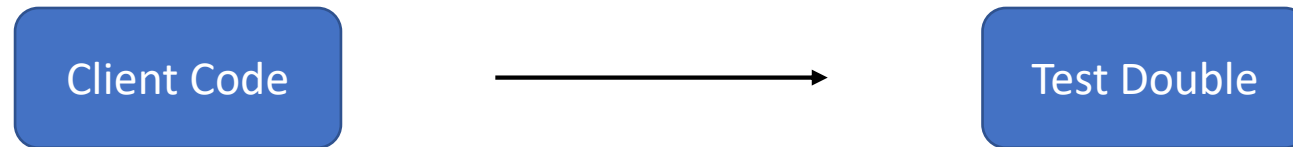
TEST DOUBLES

On Every Continuous Integration Run

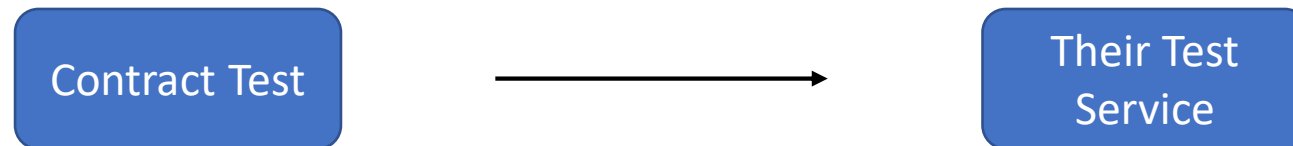


CONTRACT TESTING

On Every Continuous Integration Run



Occasionally



CONSUMER DRIVEN CONTRACTS

- Provider runs tests which verify they won't break consumers
- Shifts responsibility to the provider
- Acts as a communication method

TESTING WEB APPS

- Adding waits to tests makes them non-deterministic (flaky)
- Non-deterministic tests are bad

TESTING WEB APPS

- Test specifications in markdown
- Multiple language support
- Maintainable tests
- GPL v3.0
- Reliable browser automation
- Request/Response stubbing and mocking
- Smart Selectors
- MIT License



<https://gauge.org/>

TAIKO

<https://taiko.gauge.org/>

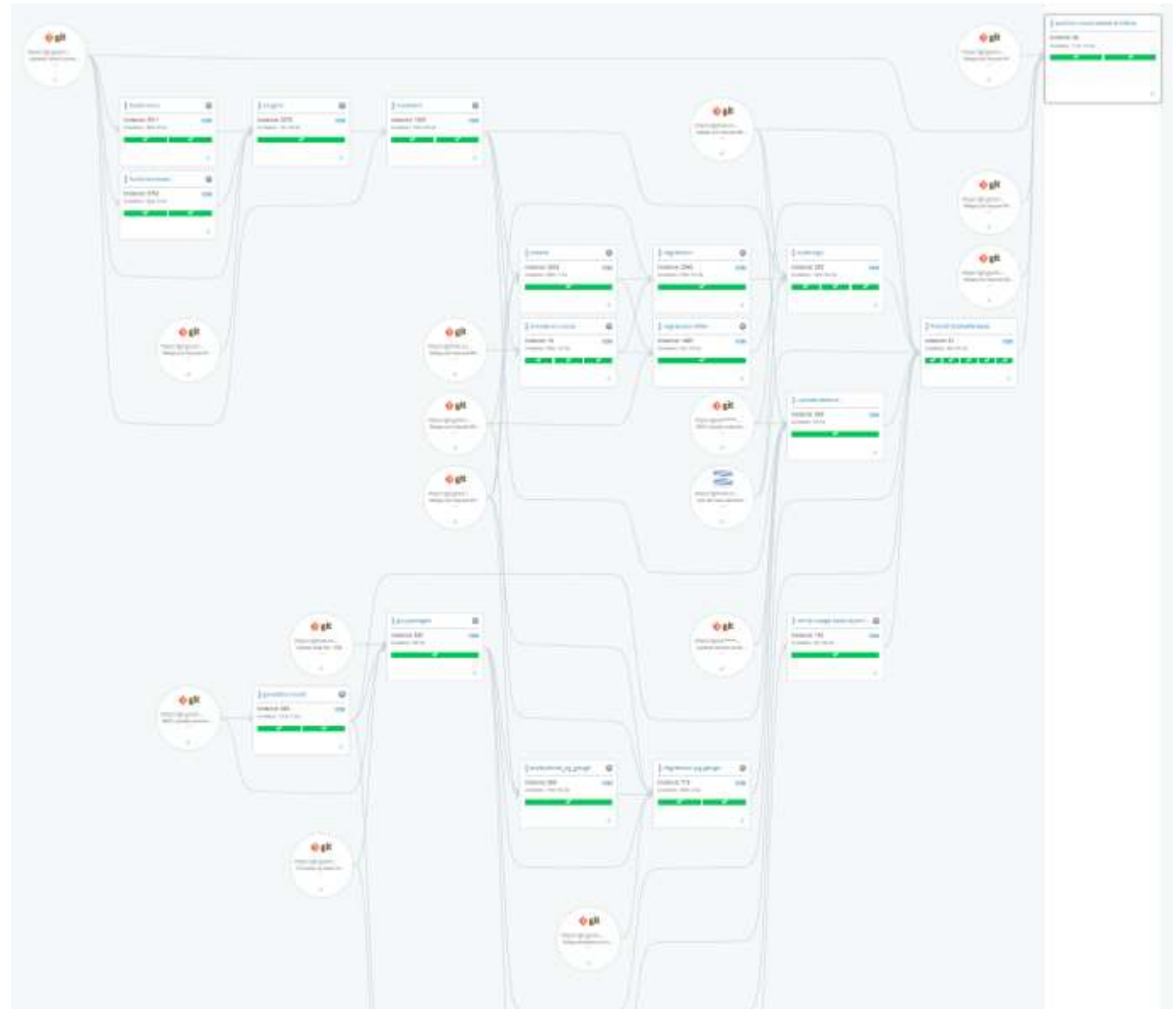
OBSERVABILITY

In control theory, observability is a measure of how well internal states of a system can be inferred from knowledge of its external outputs. The observability and controllability of a system are mathematical duals. The concept of observability was introduced by Hungarian-American engineer Rudolf E. Kálmán for linear dynamic systems.

DOMAIN ORIENTED OBSERVABILITY

```
applyDiscountCode (discountCode) {  
    this.instrumentation.applyingDiscountCode (discountCode) ;  
  
    let discount;  
    try {  
        discount = this.discountService.lookupDiscount (discountCode) ;  
    } catch (error) {  
        this.instrumentation.discountCodeLookupFailed (discountCode, error) ;  
        return 0 ;  
    }  
    this.instrumentation.discountCodeLookupSucceeded (discountCode) ;  
  
    const amountDiscounted = discount.applyToCart (this) ;  
    this.instrumentation.discountApplied (discount, amountDiscounted) ;  
    return amountDiscounted ;  
}
```

OUR ACTUAL PIPELINE



DEPLOYMENT

EXPAND / CONTRACT FOR DATABASES

DB Version 1

- ID (not null)
- Name (not null)
- Email (not null)

DB Version 2

- ID (not null)
- Name (not null)
- FirstName
- LastName
- Email (not null)

DB Version 2

- ID (not null)
- Name (not null)
- FirstName
- LastName
- Email (not null)

App Version 1

App Version 1

App Version 2

EXPAND / CONTRACT FOR DATABASES

DB Version 2

- ID (not null)
- Name (not null)
- FirstName
- LastName
- Email (not null)

App Version 2

```
select FirstName, LastName from users
```

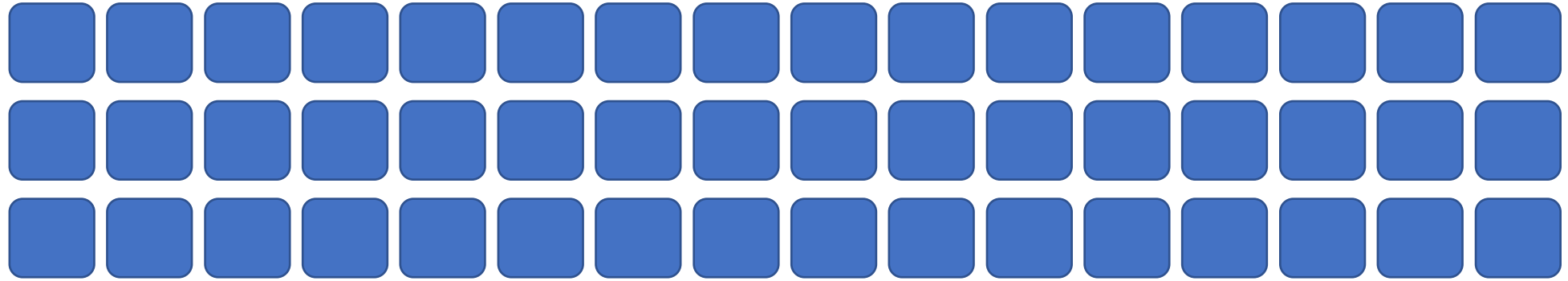
```
If null select Name from users
```

```
  Prompt user to update
```

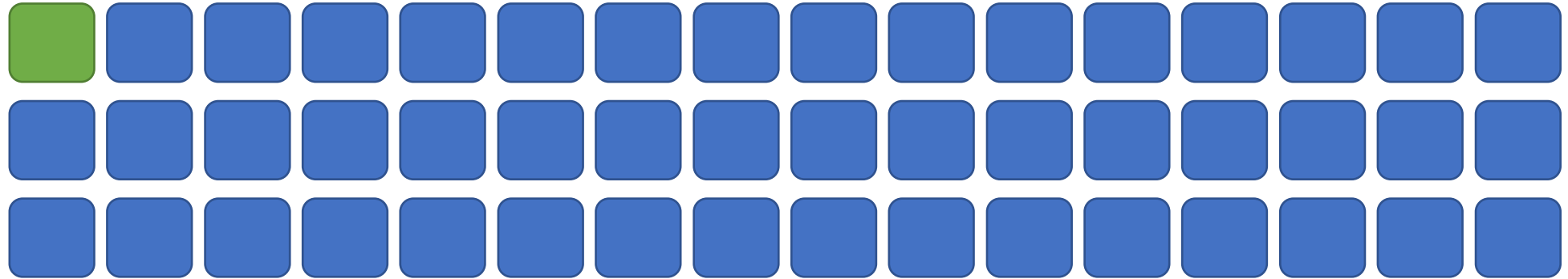
```
  Update FirstName, LastName
```

```
select FirstName, LastName from users
```

CANARY RELEASE



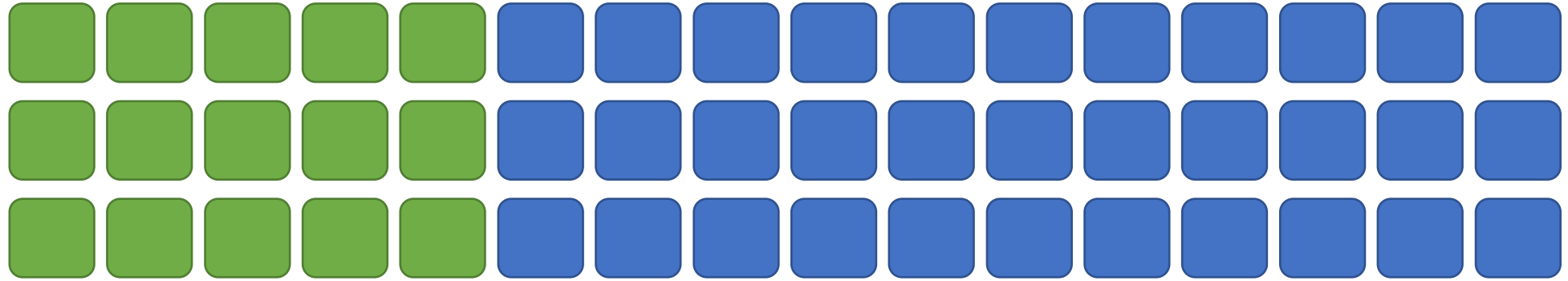
CANARY RELEASE



Release to a very small portion of our audience to “test in production”.

(Pro Tip: Test for business efficacy not just successful bits and bytes)

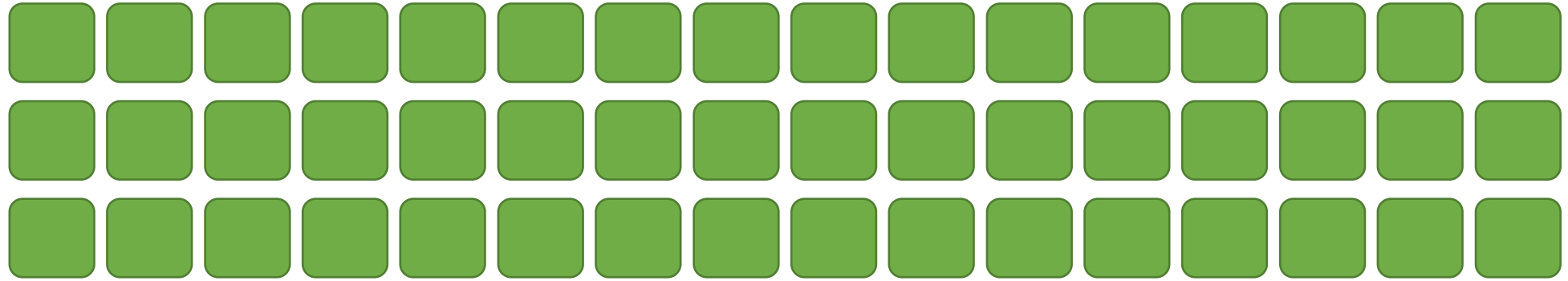
CANARY RELEASE



Release to a very small portion of our audience to “test in production”.

(Pro Tip: Test for business efficacy not just successful bits and bytes)

CANARY RELEASE



Release to a very small portion of our audience to “test in production”.

(Pro Tip: Test for business efficacy not just successful bits and bytes)

S U M M A R Y

- Small, well versioned pieces are best
- Small, well focused teams are best to create those
- Continuous Integration is the foundation

THANK YOU