



What they don't tell you about μ -services...

QCon NY - June 2016

Daniel Rolnick

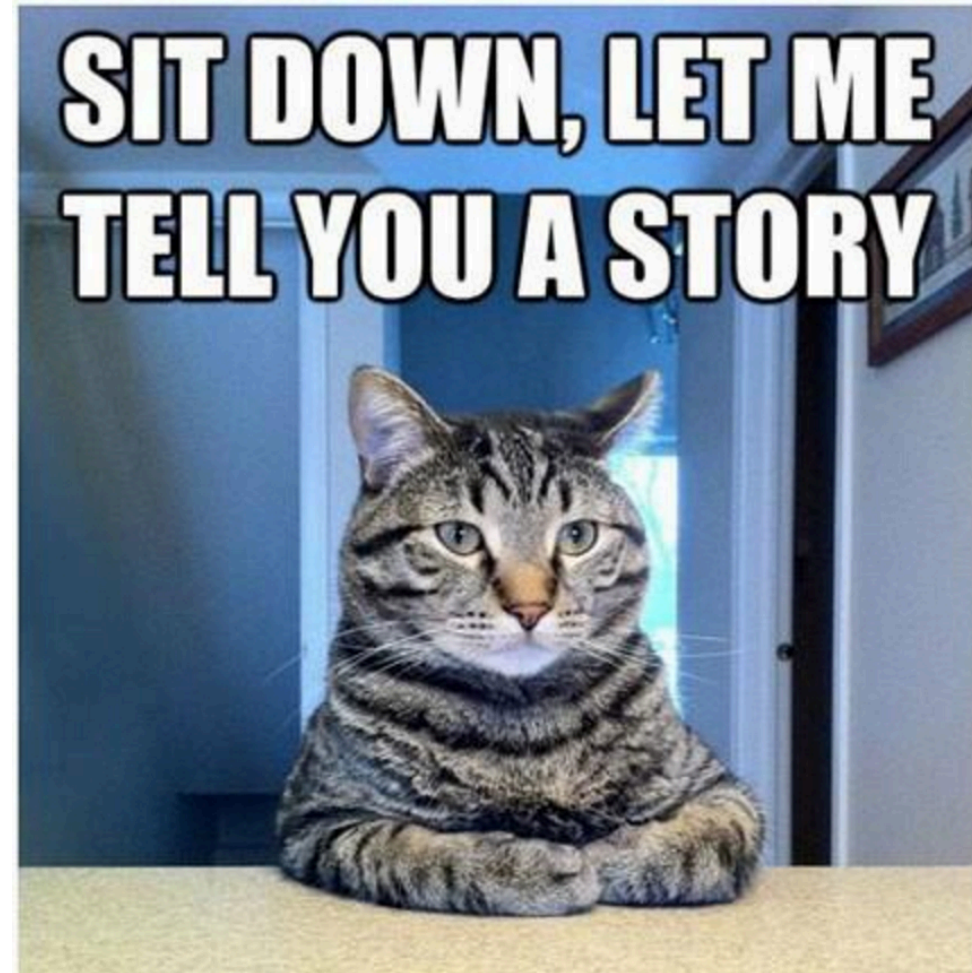
Chief Technology Officer

Daniel Rolnick

Chief Technology Officer

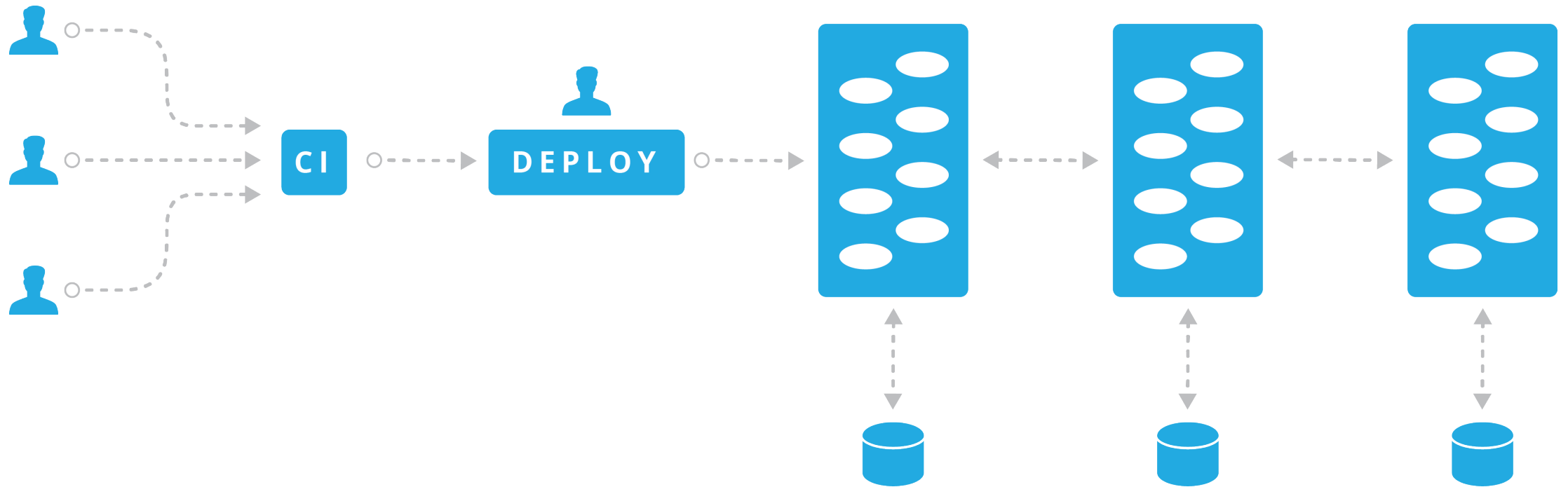
daniel.rolnick@yodle.com

Story Time



Story Time

September 2014

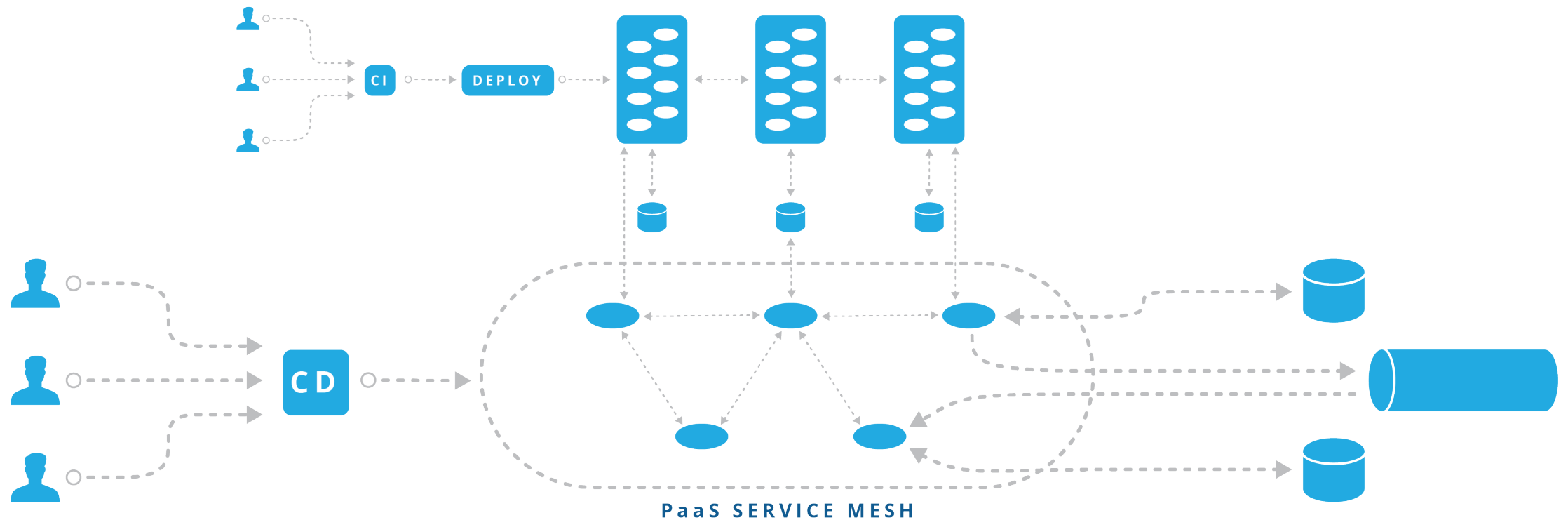


Story Time

June 2016



MONITORING



Evolution Requires Adaptation

Something's gotta give

- ▶ Changing environments cause stress
- ▶ Existing processes need to be revisited
- ▶ Processes need to be created
- ▶ New technology needs to be integrated
- ▶ Businesses are built on trade-offs



Eyes Wide Open

Expected developmental needs

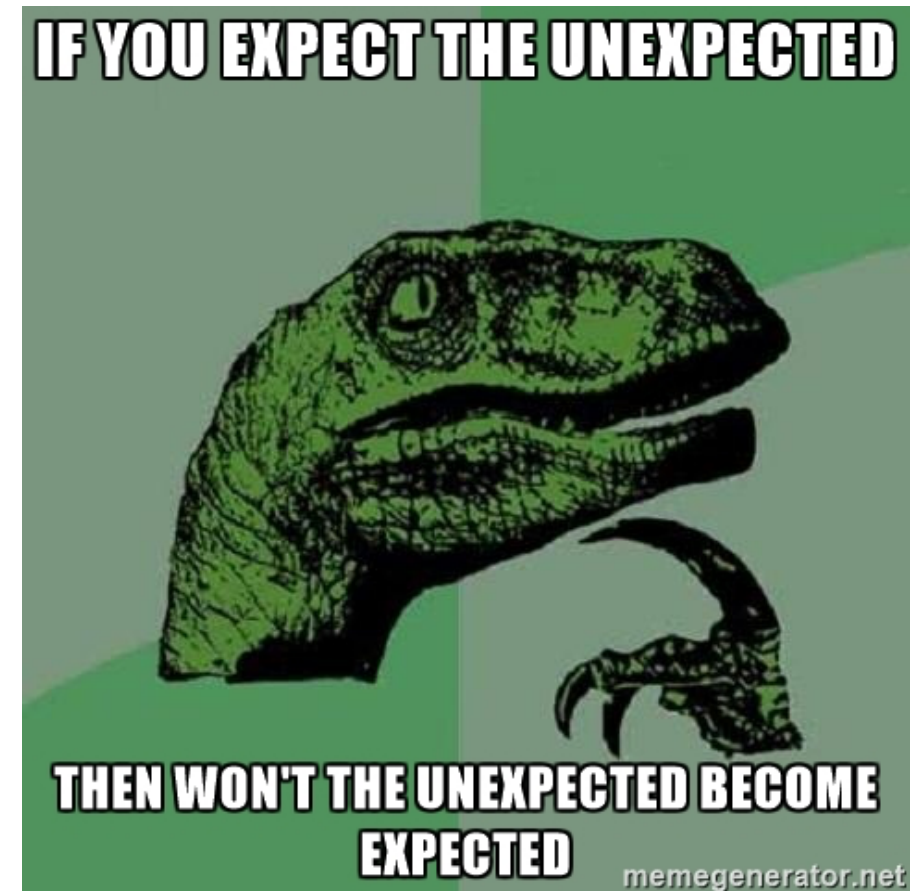
- ▶ Platform as a Service
- ▶ Service Discovery
- ▶ Testing
- ▶ Containerization
- ▶ Monitoring



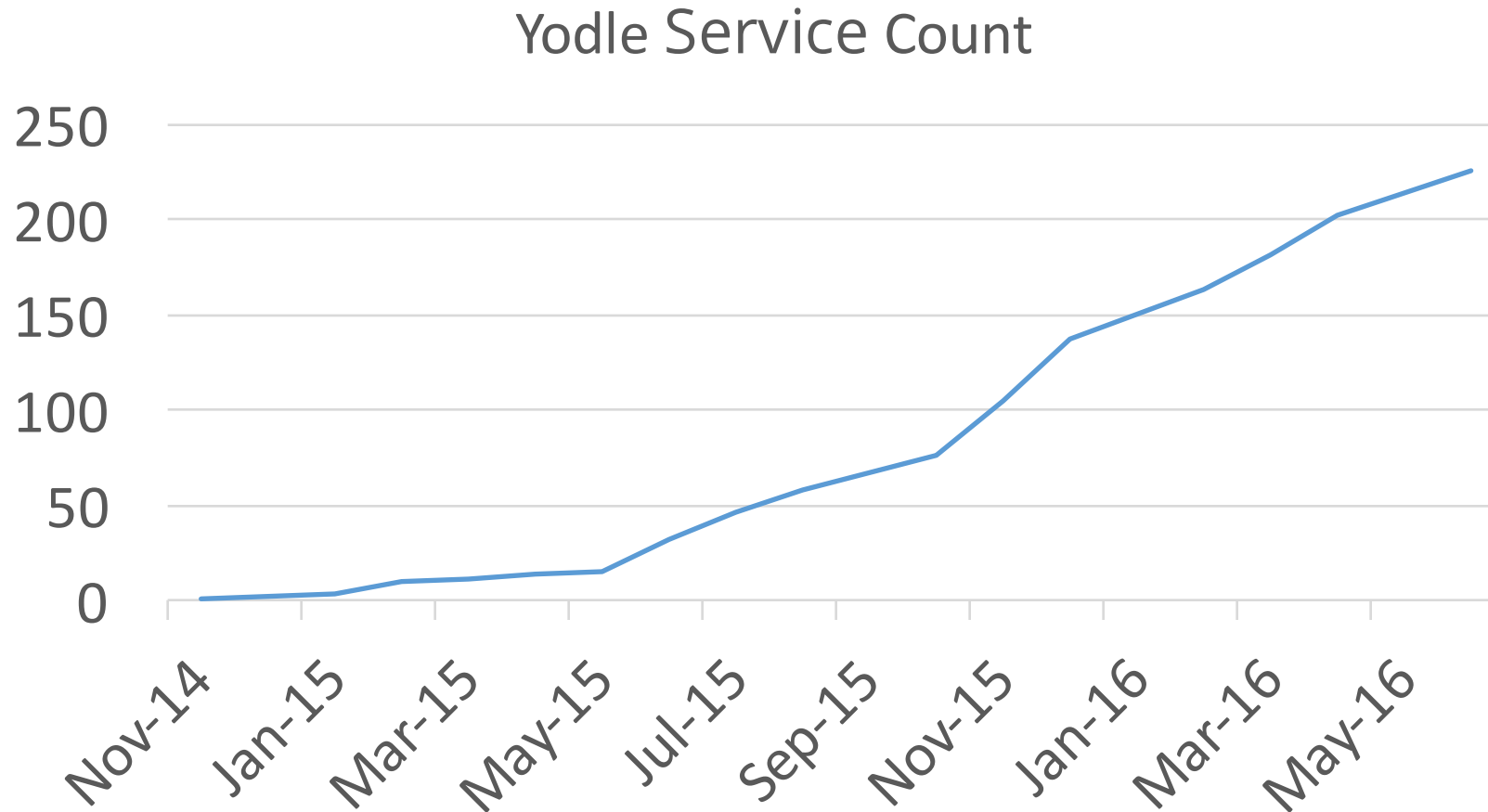
Expect the Unexpected

Unexpected implications of micro-services

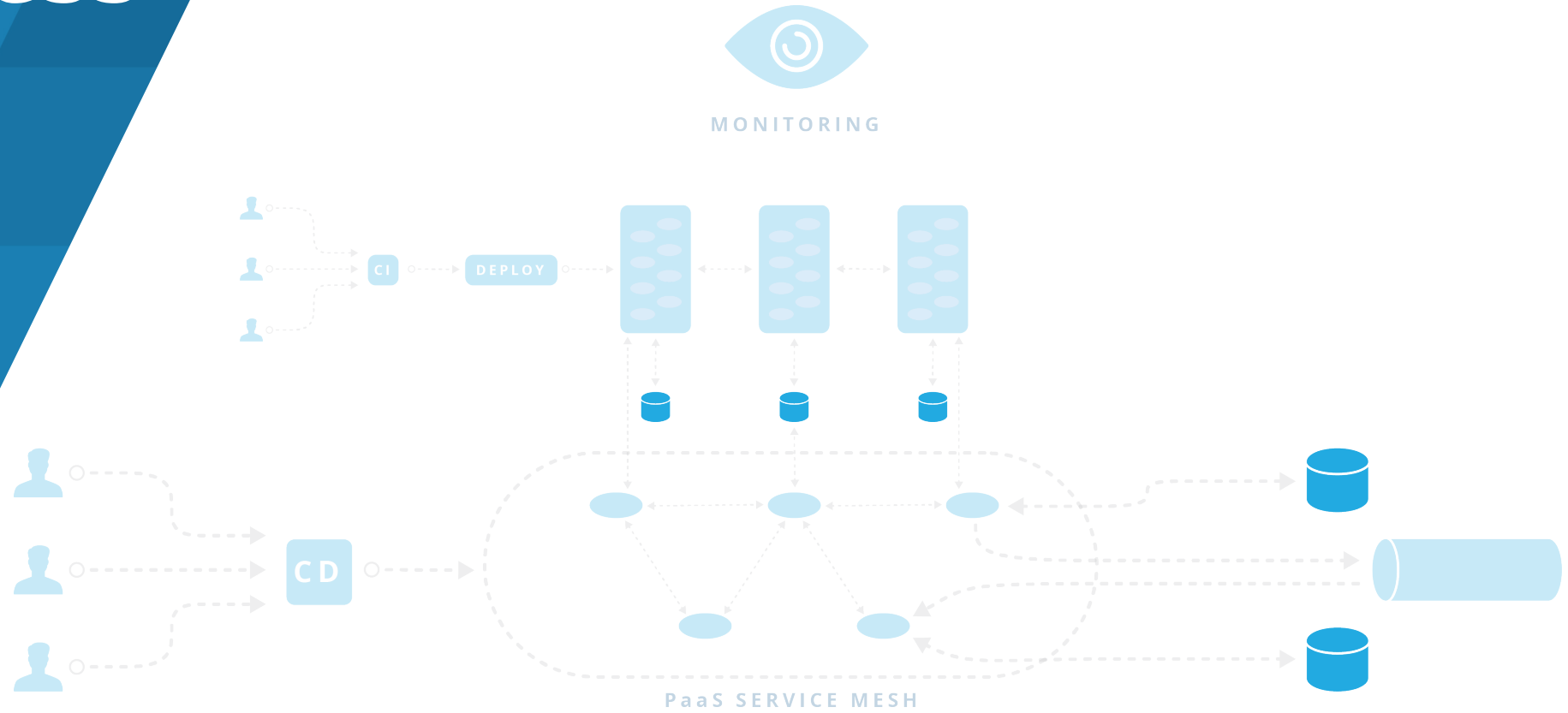
- ▶ Impact on data access
- ▶ Build and Deploy Tooling
- ▶ Source Repository Complexity
- ▶ Cross application monitoring



Bring on the complexity



Data access patterns



Microservices Macroproblems

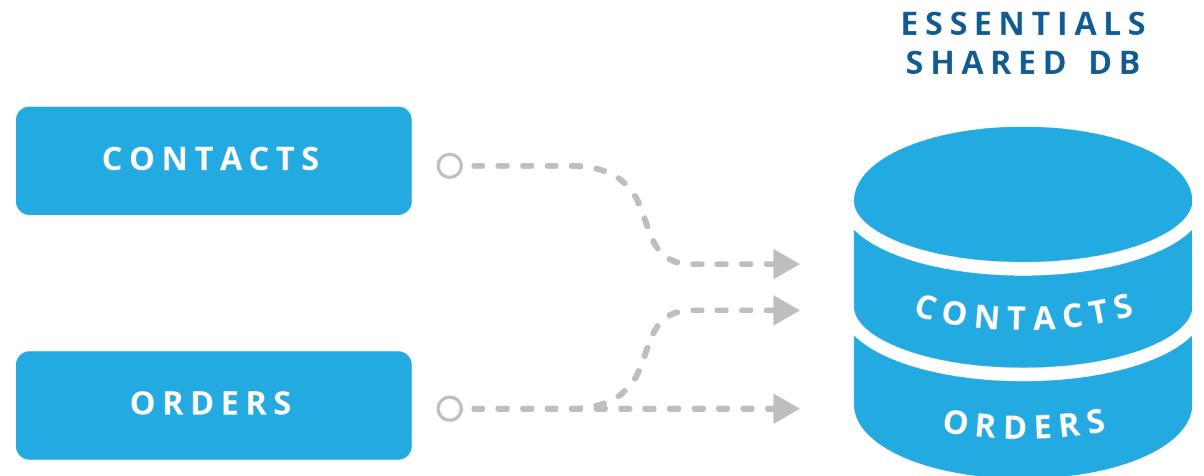
Independent Data Domains

- ▶ Isolated data ownership per micro-service
- ▶ Options: Physical Databases, Schemas, Polyglot
- ▶ Ideal state for new things but what about the old stuff
- ▶ Can't get there in one move

Microservices Macroproblems

Baby Steps to Freedom

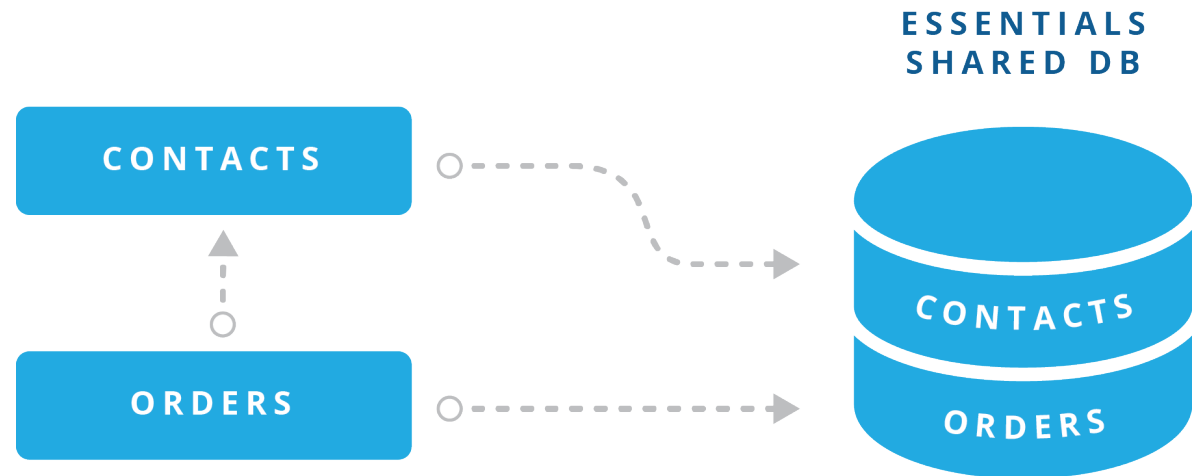
- ▶ Central data stores are leaky abstractions



Microservices Macroproblems

Baby Steps to Freedom

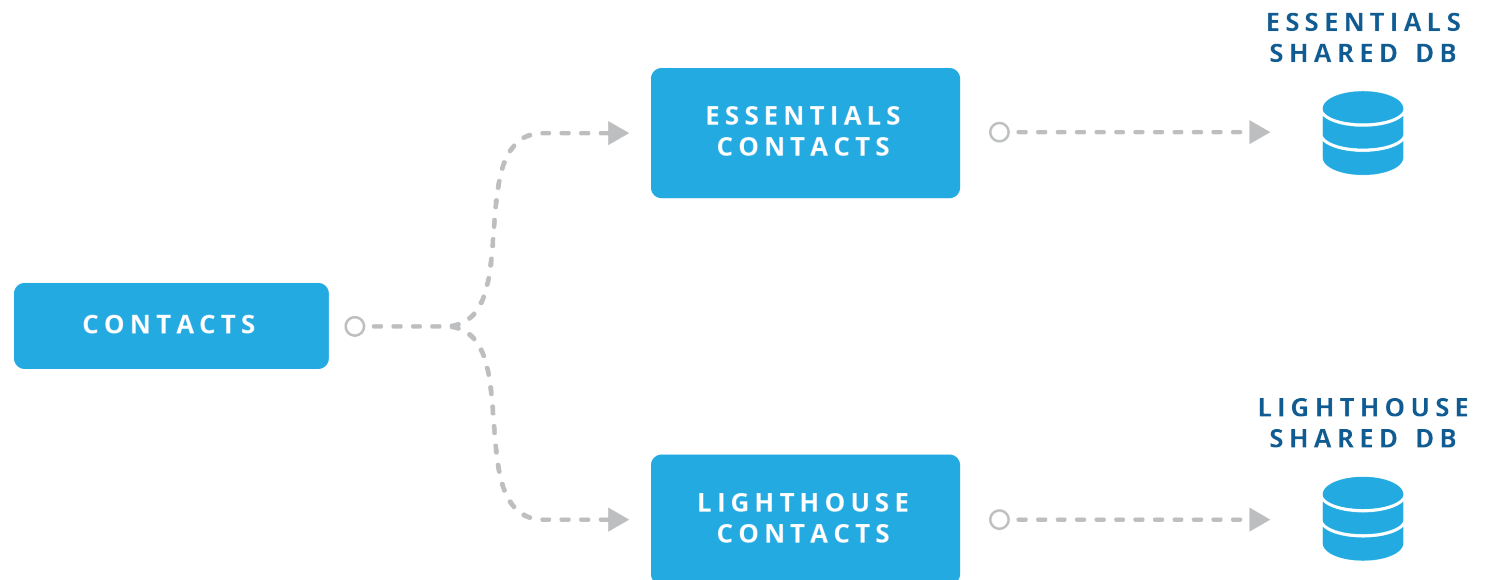
- ▶ Central data stores are leaky abstractions
- ▶ Enforce data ownership through access patterns



Microservices Macroproblems

Baby Steps to Freedom

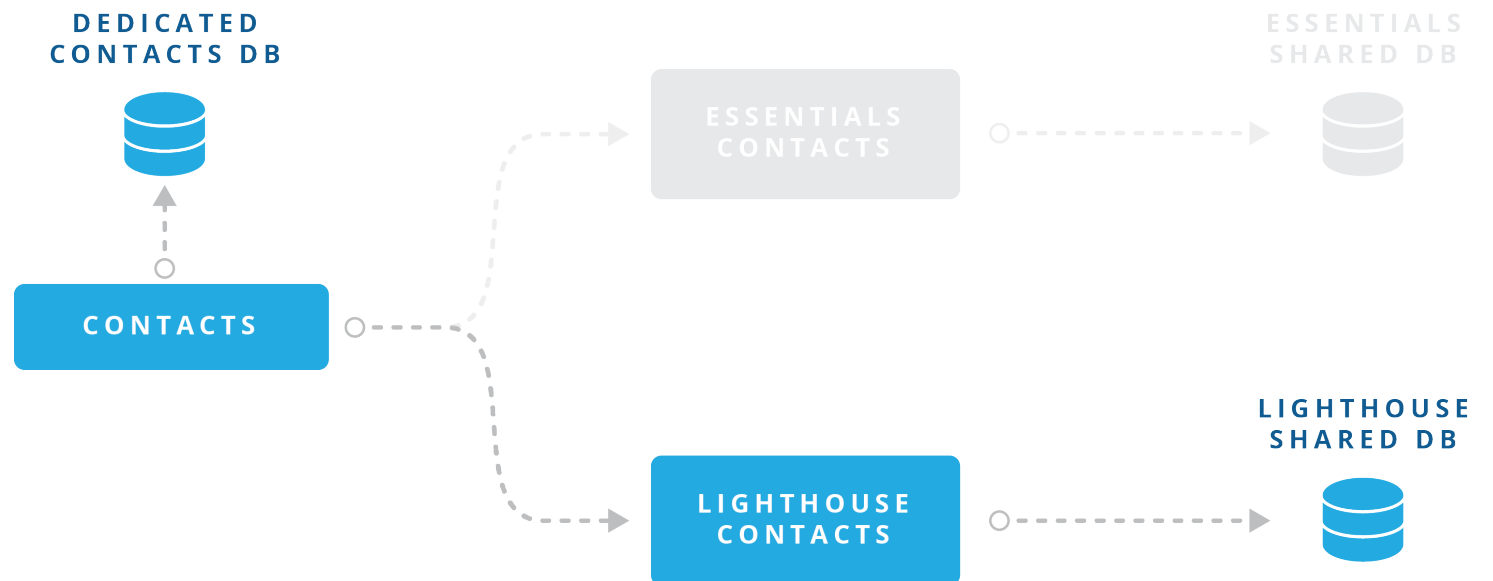
- ▶ Central data stores are leaky abstractions
- ▶ Enforce data ownership through access patterns
- ▶ Façade for decoupling



Microservices Macroproblems

Baby Steps to Freedom

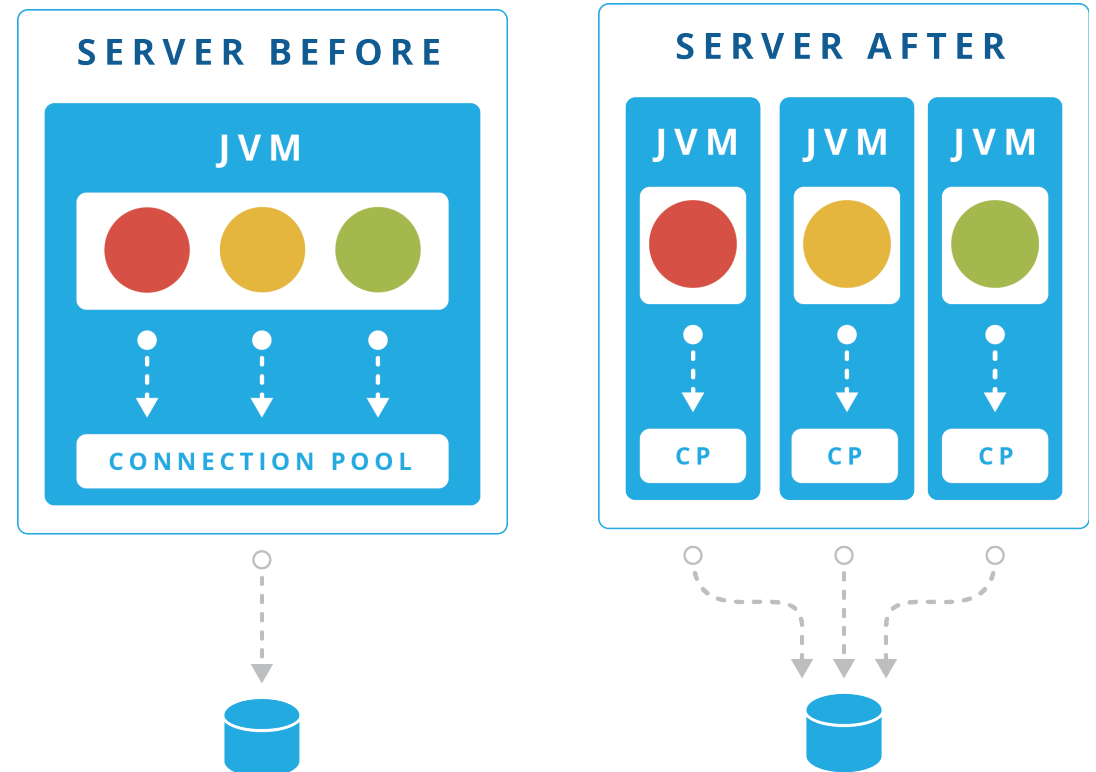
- ▶ Central data stores are leaky abstractions
- ▶ Enforce data ownership through access patterns
- ▶ Façade for decoupling
- ▶ Multi-step process



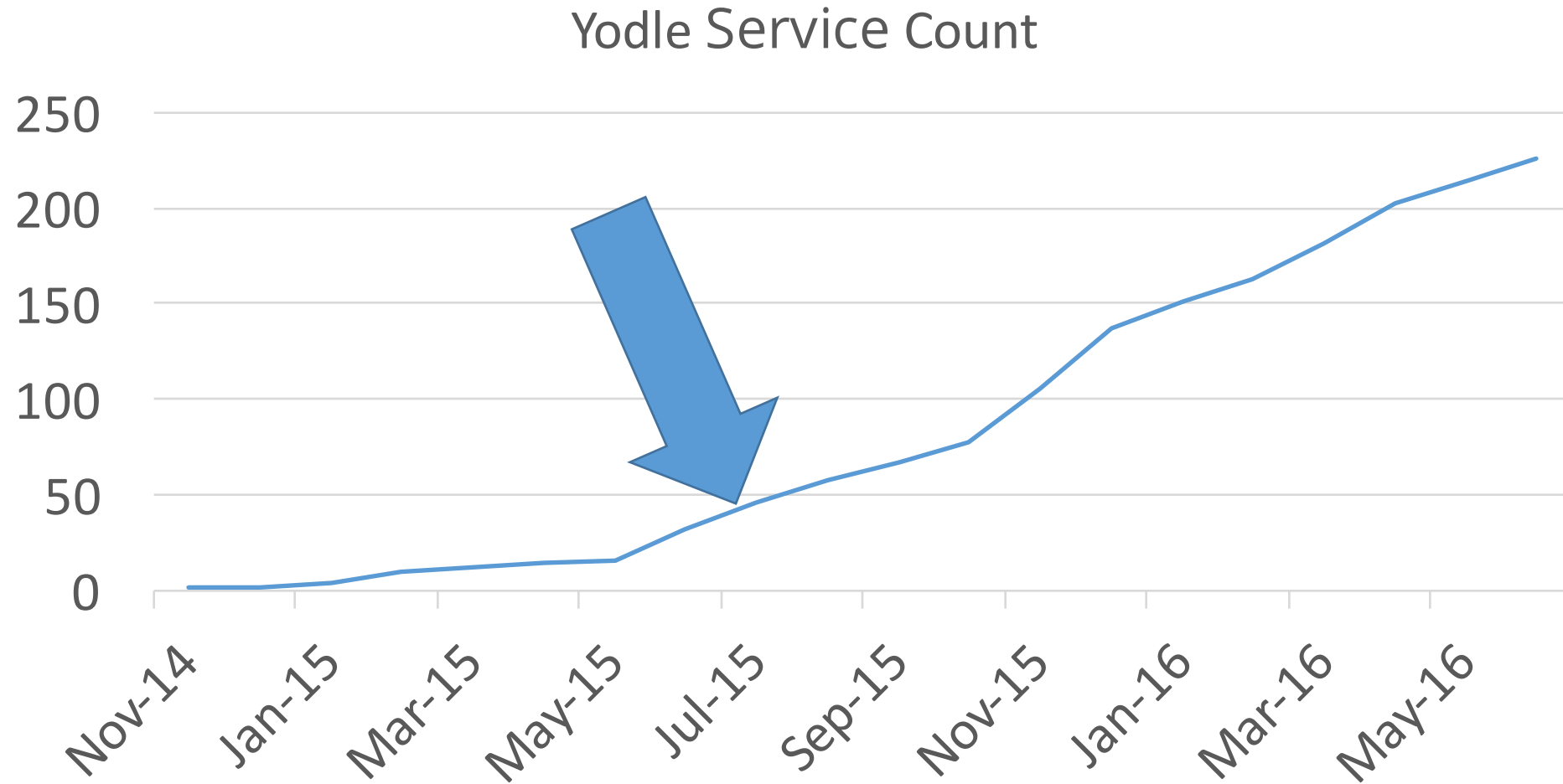
Microservices Macroproblems

Shared Containers Simplify Things

- ▶ Services in the same container reuse connections
- ▶ Connection pooling goes away
- ▶ Base connection count starts adding up
- ▶ You could always go to a minimum idle of zero
- ▶ What could go wrong?



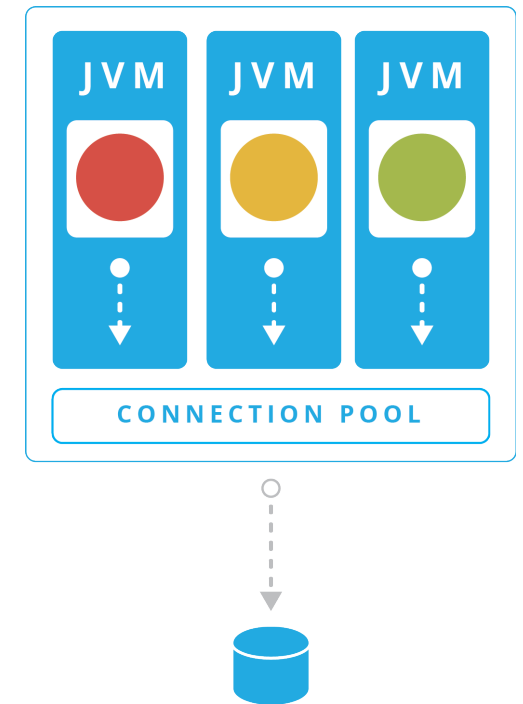
Microservices Macroproblems



Microservices Macroproblems

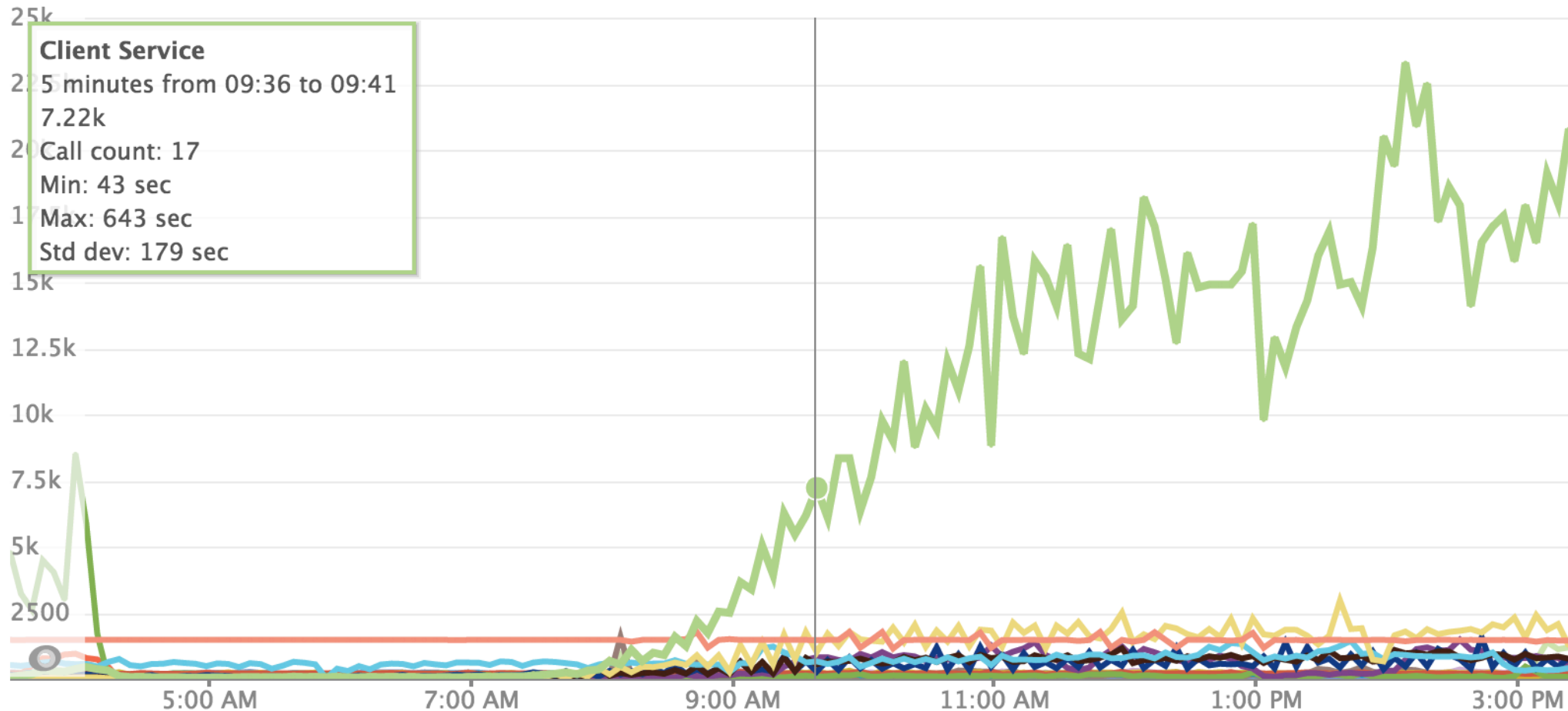
External Connection Pooling

- ▶ Connection pooling outside of the container
- ▶ Add visibility while you're at it
- ▶ Better logging, cleaner visualizations



Microservices Macroproblems

Connections Opened Per Application



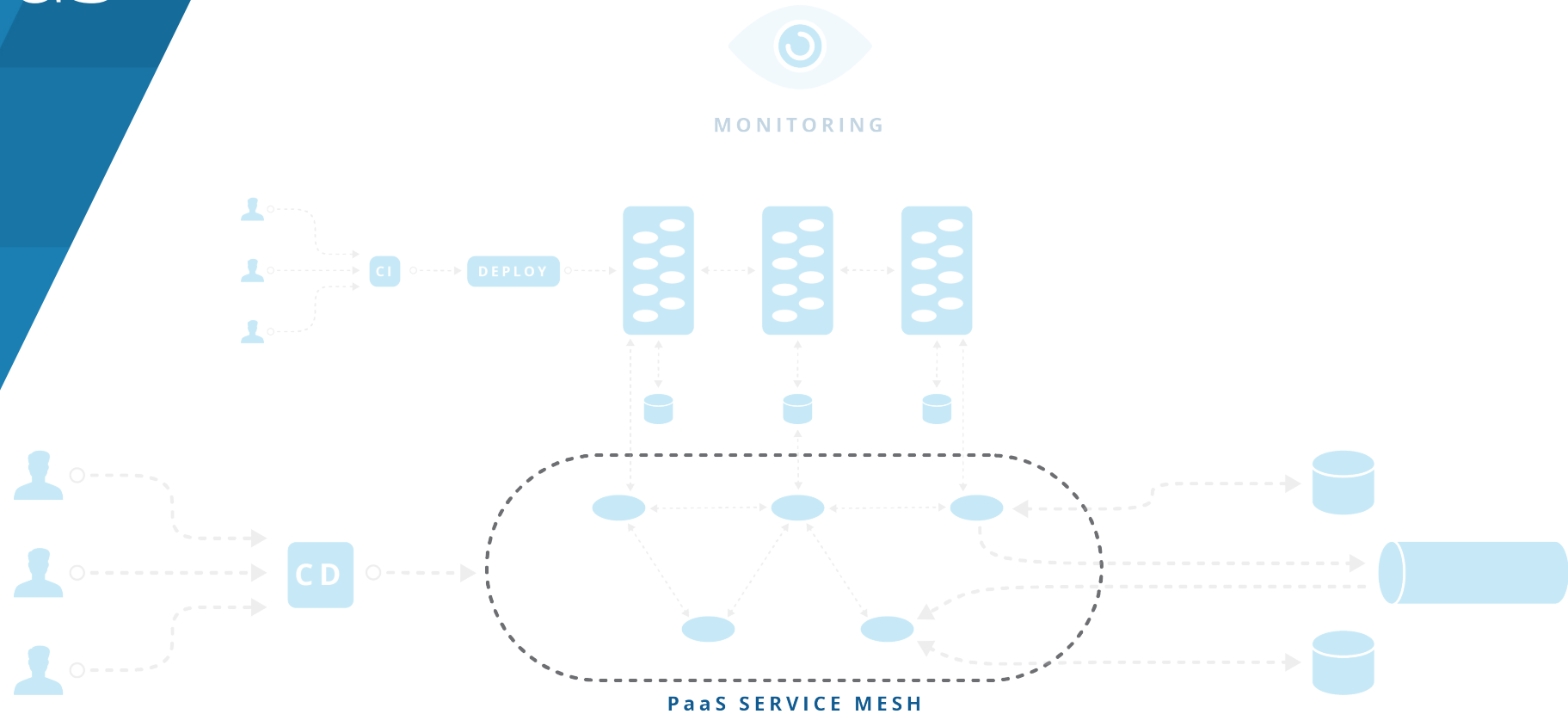
Microservices Macroproblems

Tooling for empowerment

- ▶ Server spin-up
- ▶ Schema and Account creation
- ▶ Ensure externalized your configurations



Platform as a Service



A Place for Everything and Everything...

Static Configurations

- ▶ Every application deployed to a fixed set of hosts on a set of known ports
- ▶ Monitoring was done at a gross system synthetic level
- ▶ Only complete outages were easily detectable
- ▶ Manual restarts required
- ▶ PS-Watcher and Docker restart help but are not sufficient
- ▶ This was not going to scale

This Ain't Gonna Scale

Keeping services alive by hand is problematic

- ▶ Researched available PaaS Platforms available in late 2014
 - Mesos / Marathon
 - CoreOS
- ▶ What about:
 - Kubernetes
 - Swarm
 - AWS Elastic Container Service

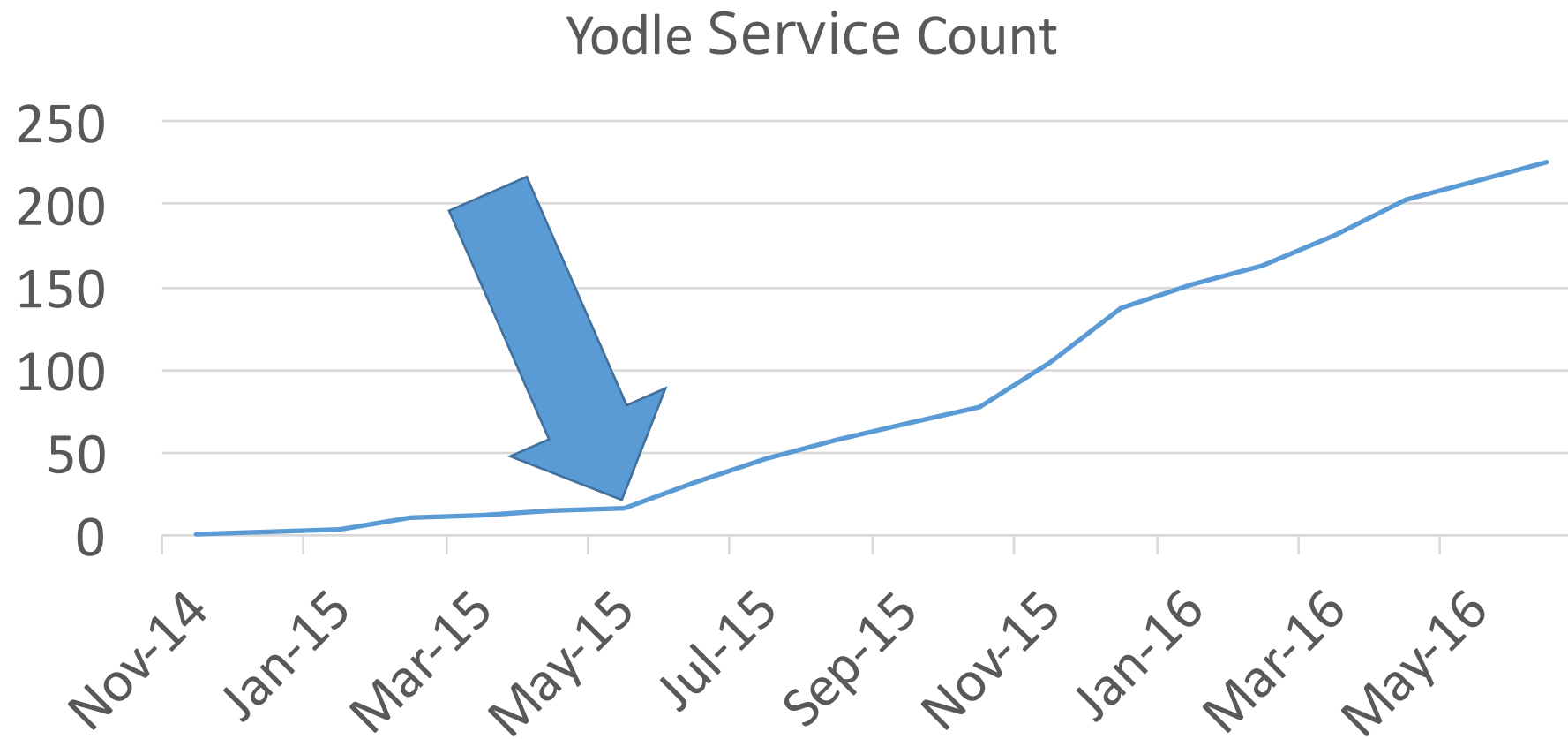
Platform as a Service

Mesos and Marathon

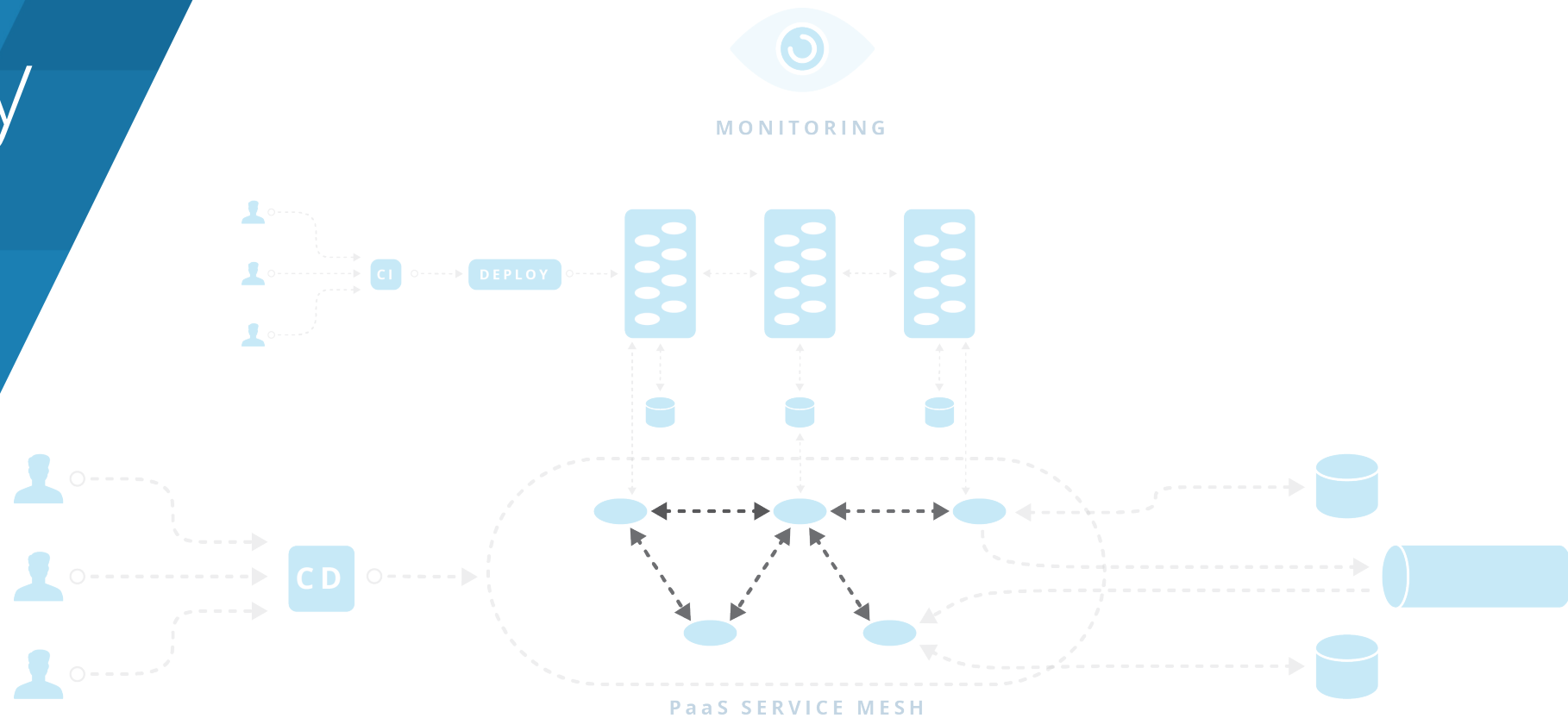
- ▶ Deploy applications to marathon
- ▶ Marathon decides what host and port to run applications on
- ▶ Health checks are built in to ensure application up-time
- ▶ Mesos ensures the applications run and are contained

Platform as a Service

Pace of Innovation Increases



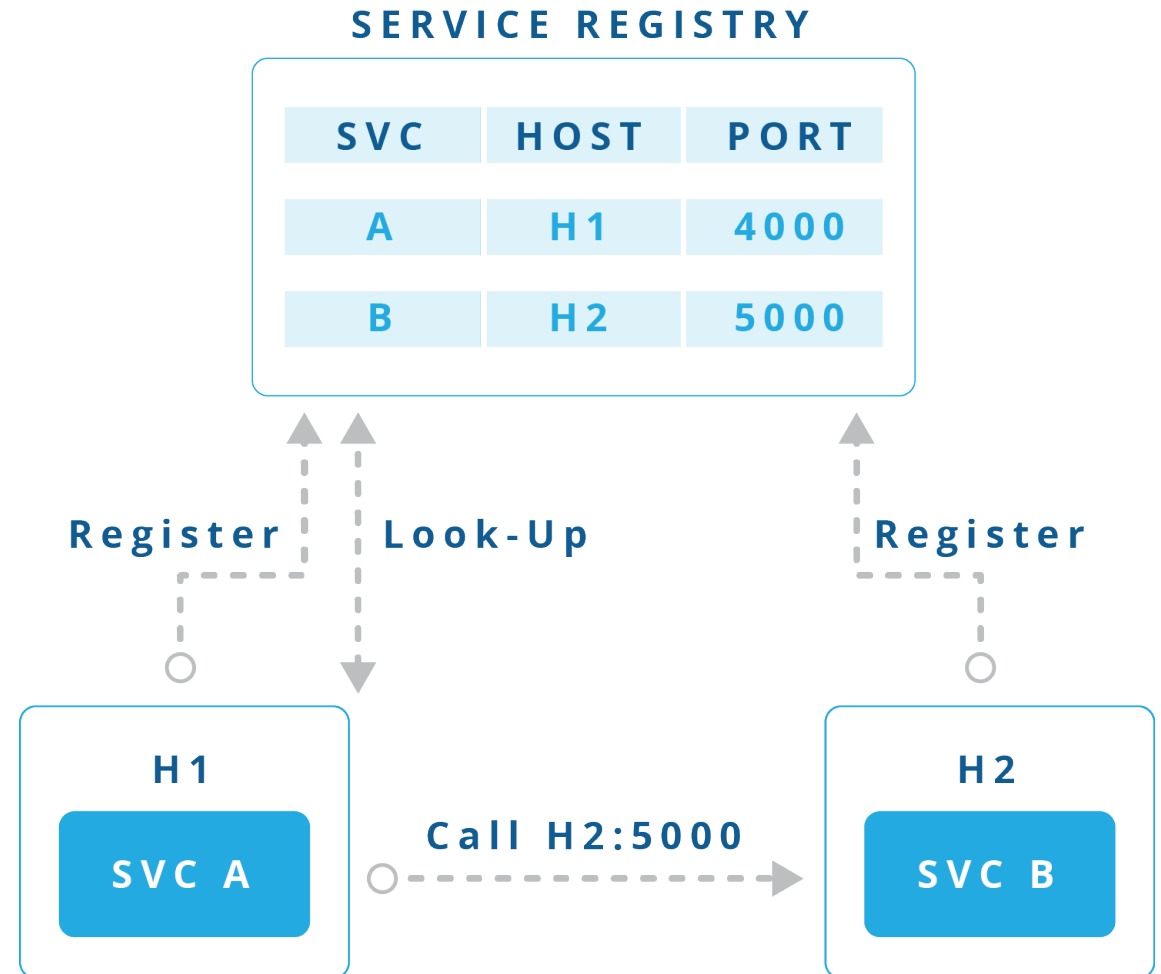
Service Discovery



Dynamic Topologies Require Service Discovery

Aware Apps vs. Smart Pipes

- ▶ Service discovery can be baked into your application



Dynamic Topologies Require Service Discovery

Aware Apps vs. Smart Pipes

- ▶ Plumbing can take care of it for you

- ▶ Smart Pipes allows
 - Easier path to polyglot ecosystem
 - Decouple applications from service discovery



- ▶ We chose the latter but we had to iterate a few times to get there

Use What You Know

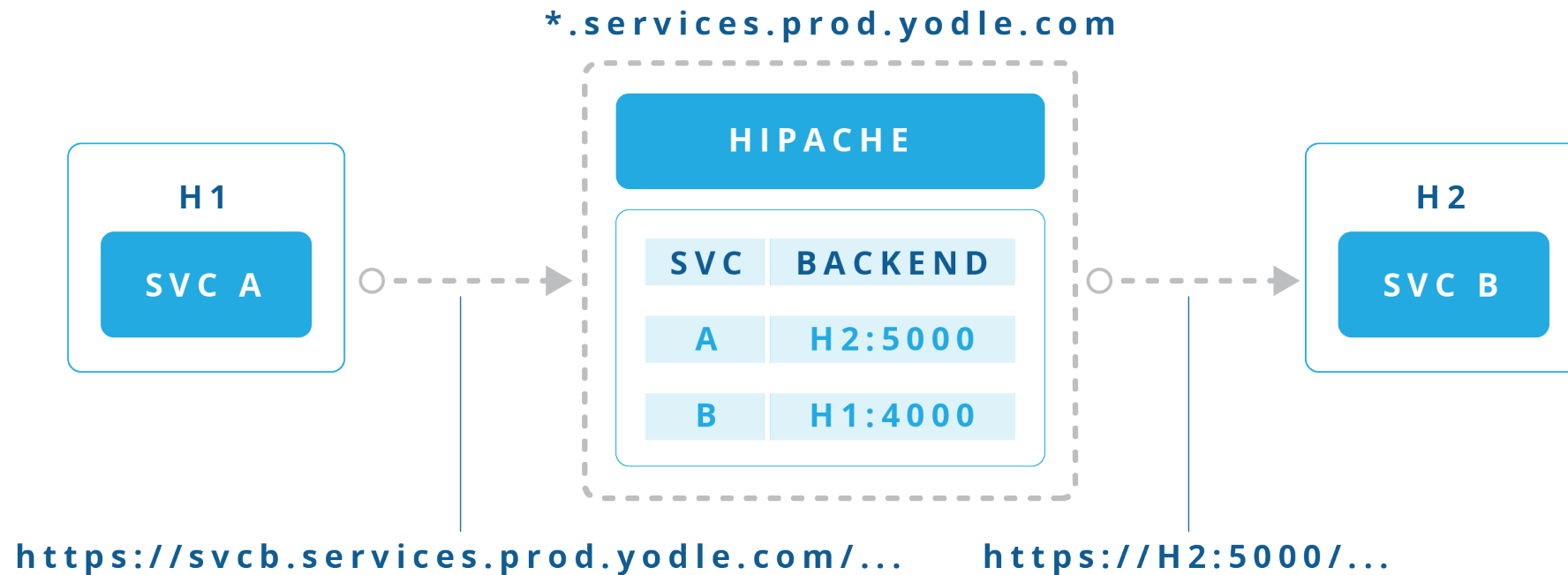
Curator already in place

- ▶ Already used zookeeper/curator for our thrift based macro-services
- ▶ Made our micro-services self register and do discovery via curator
- ▶ You can't solve everything at once
- ▶ Not our desired end state

Service Discovery V2

Hipache by dotCloud

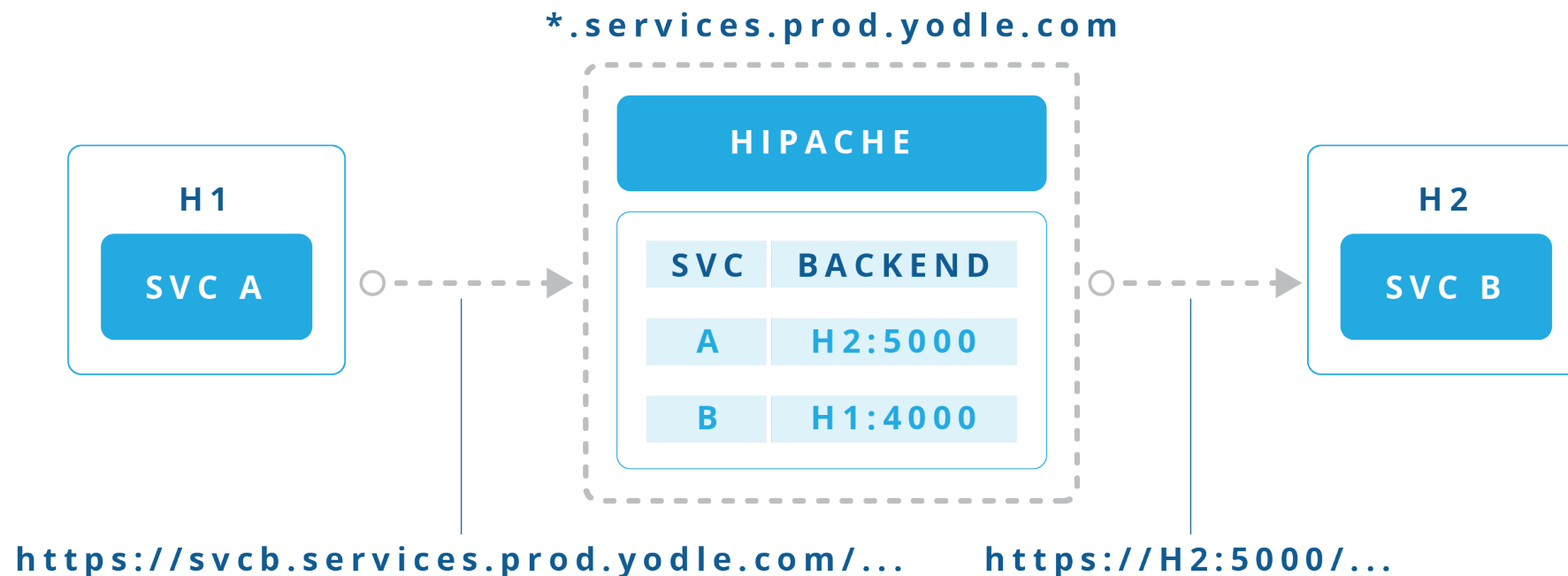
- ▶ URLs looked like <https://svcb.services.prod.yodle.com>
- ▶ Utilized dedicated routing servers



Service Discovery V2

Hipache by dotCloud

- ▶ Pros: Decoupled service discovery from applications
- ▶ Cons: Services had to be environment aware



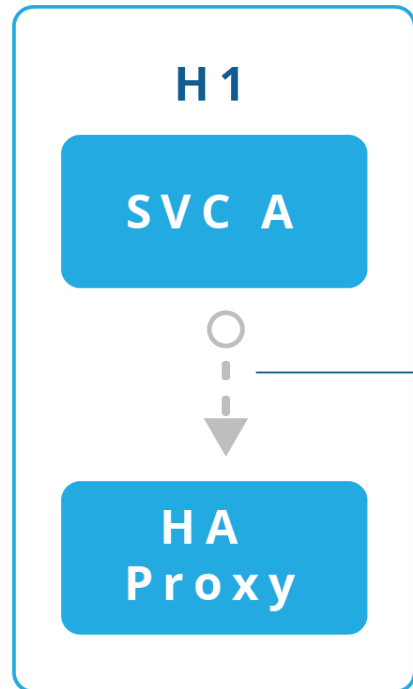
PaaS's built-in routing layer

- ▶ Marathon has a built-in routing layer using haproxy
- ▶ Simple command to generate an haproxy config
- ▶ Basic listener (Qubit Bamboo) keep haproxy files up-to-date
- ▶ Hipache could have worked

Service Discovery V3 Continued

Discovery was simpler

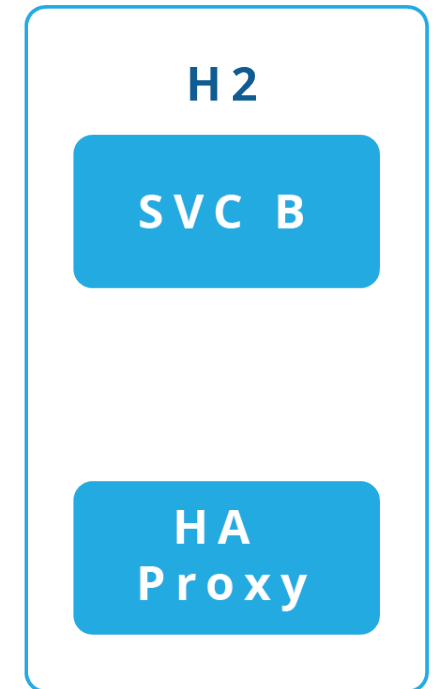
`*.services.yodle.com`



`https://svcb.services.yodle.com/...`

`https://H2:5000/...`

`*.services.yodle.com`

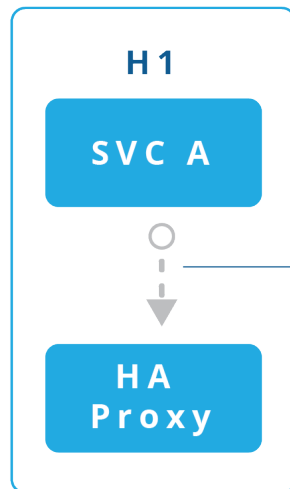


Service Discovery V3 Continued

Discovery was simpler

- ▶ Service discovery is now fully externalized
- ▶ Iterate on routing and discovery independently
- ▶ Created tech debt for the applications

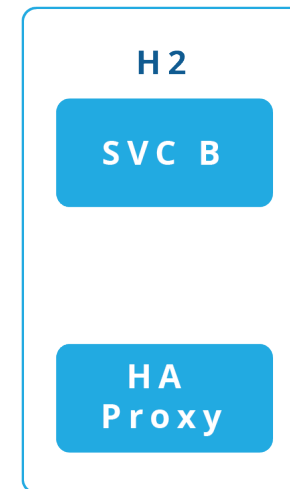
***.services.yodle.com**



`https://svcb.services.yodle.com/...`

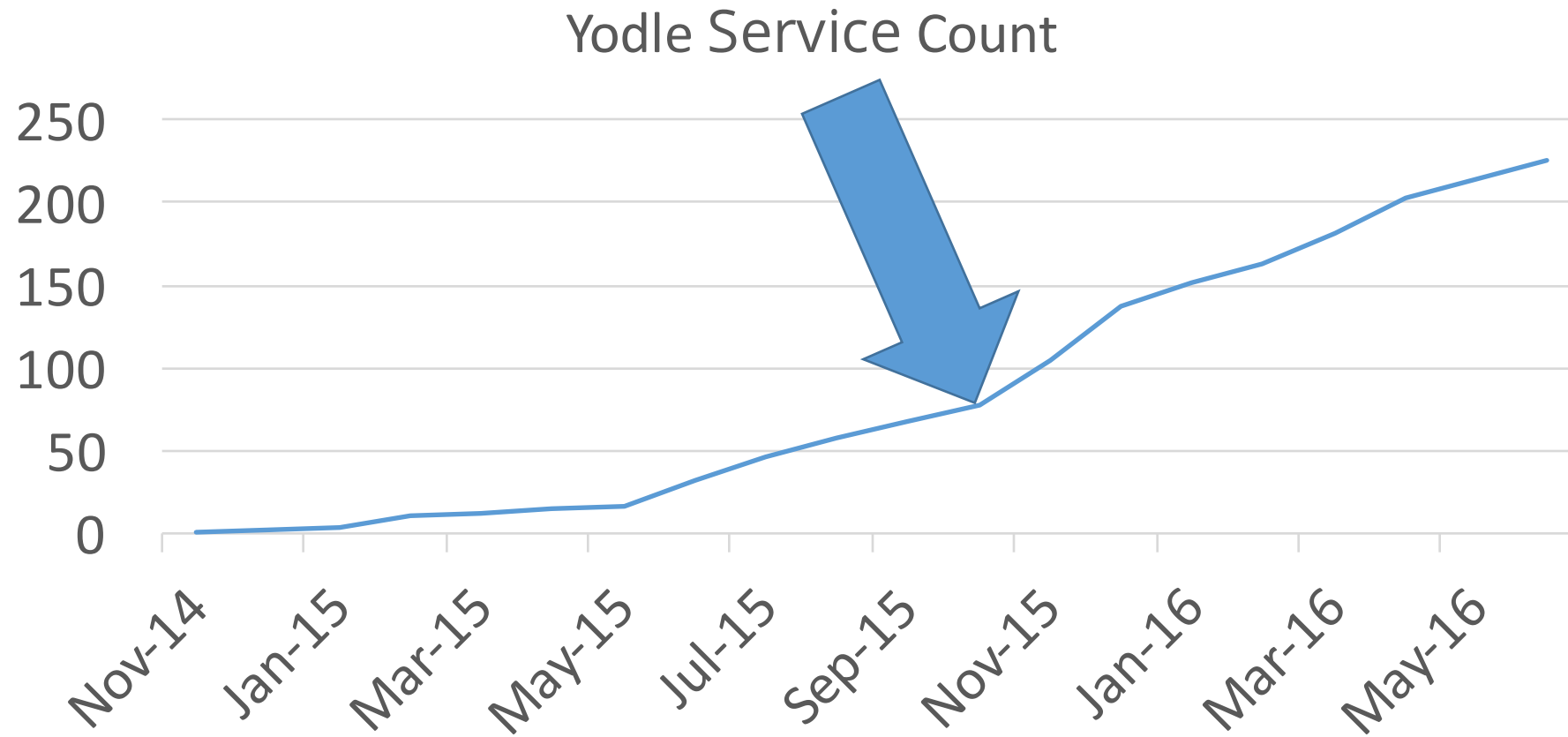
`https://H2:5000/...`

***.services.yodle.com**



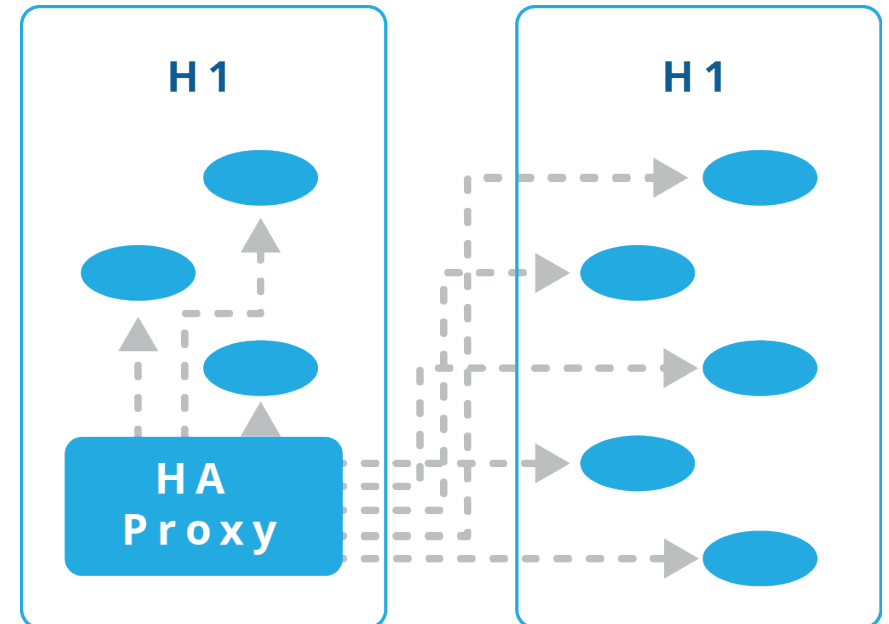
Service Discovery V4

Scale Problems

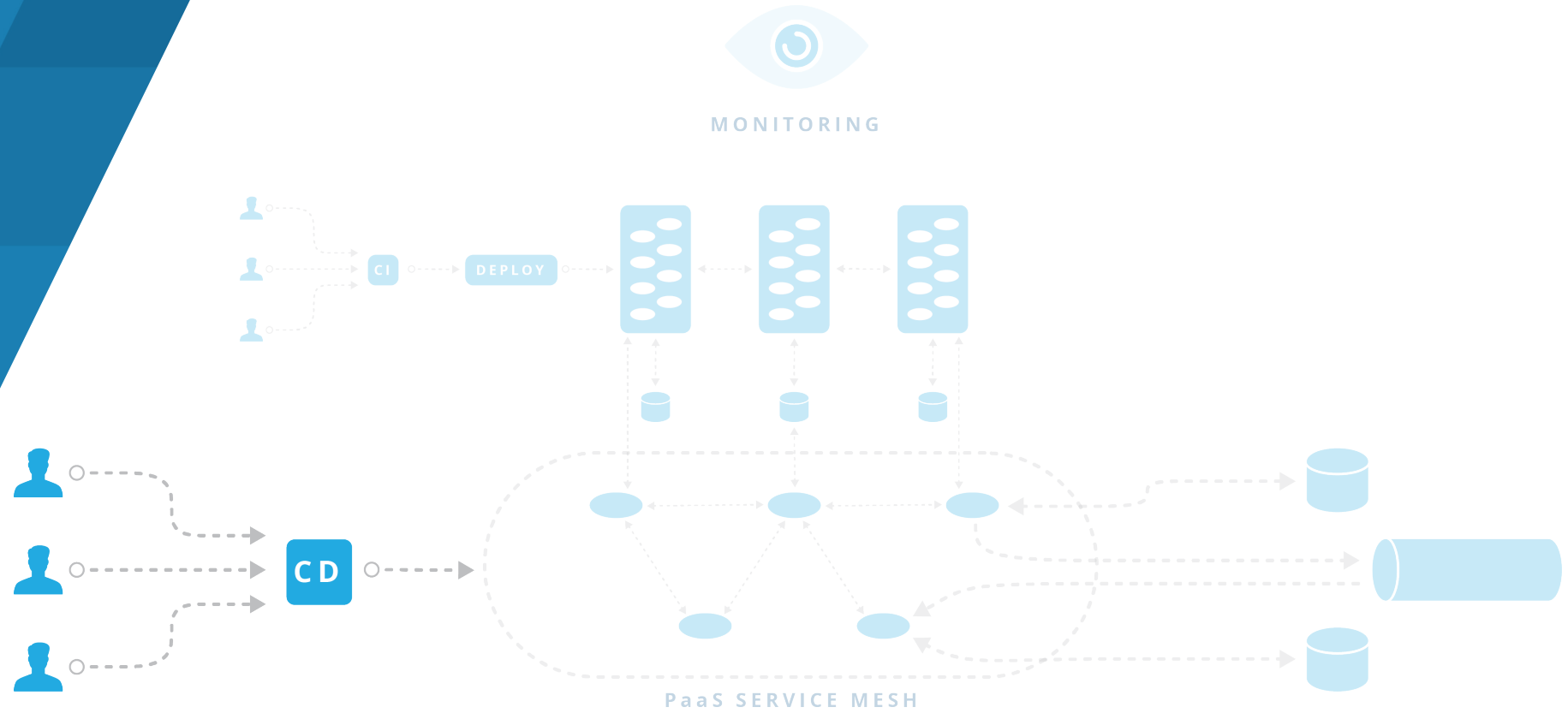


Many to Many Problems

- ▶ As the number of slave nodes in our PaaS grew so did our problems
- ▶ Health checks from every host to every container
- ▶ Ensuring the HAproxy file was up-to-date on all hosts was annoying
- ▶ Centralized onto a small cluster of routing boxes



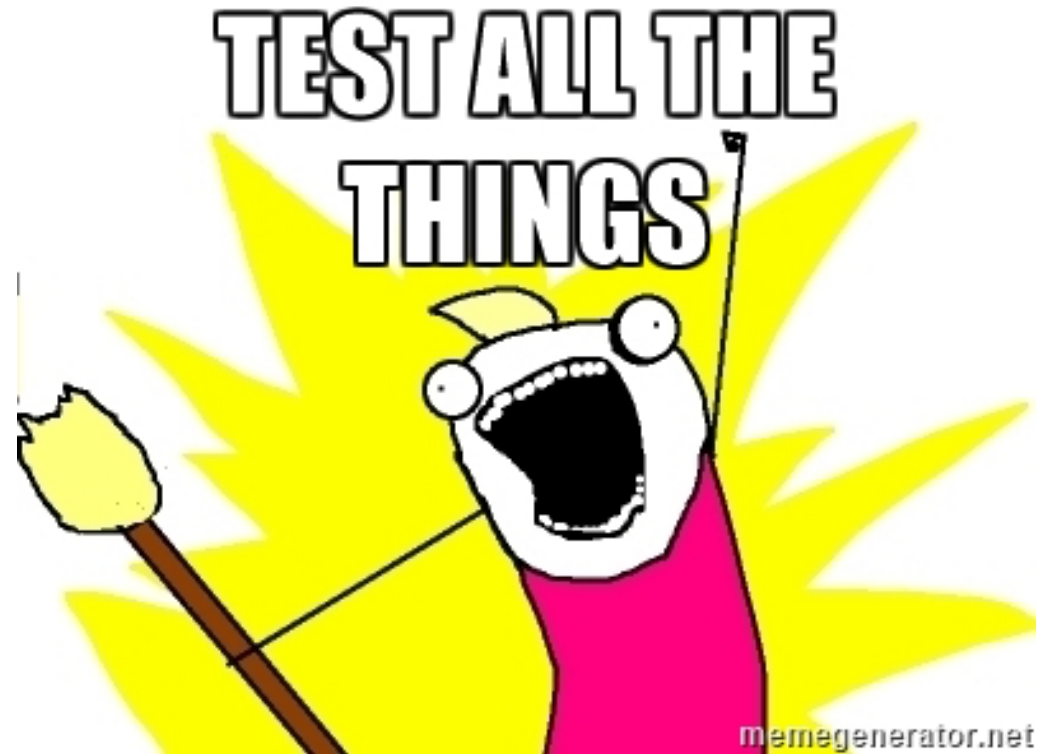
Testing



Continuous Integration

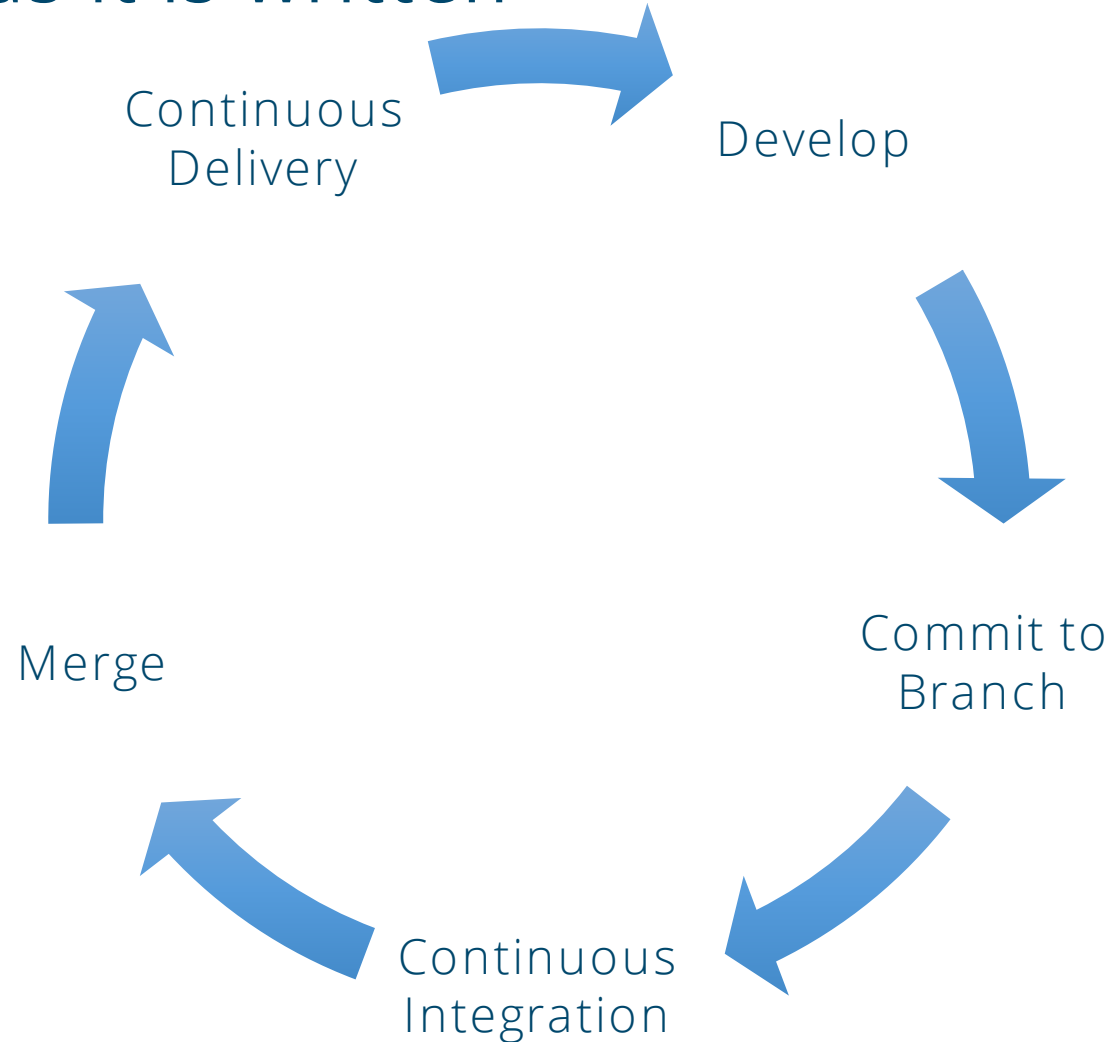
Regressions give comfort

- ▶ Monolithic releases are understandable
- ▶ We tested everything
- ▶ Everything works



Continuous Delivery Pipeline

Release code as it is written



Continuous Integration

Regressions take time

- ▶ Empower continuous delivery
- ▶ Broke apart our monolithic regression suite
- ▶ Same methodology for macro and micro-services



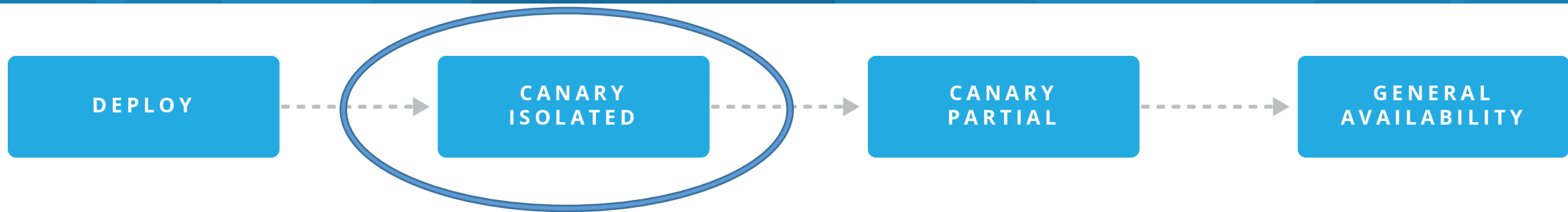
Continuous Delivery Pipeline

Enter the Canary

- ▶ Landscape is in flux
- ▶ If we test a subset of things how can we be sure everything works?
- ▶ Canary Ensures
 - ▶ Dependencies met
 - ▶ Satisfying existing contracts
 - ▶ Handle production load

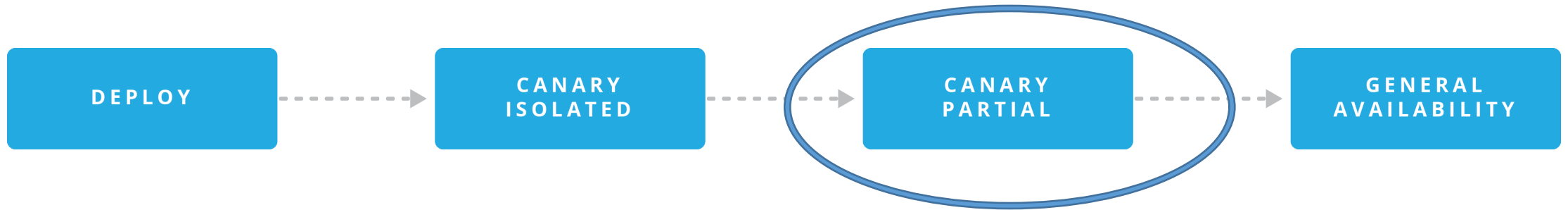


Continuous Delivery Pipeline



- ▶ Special canary routing in our service discovery layer
- ▶ Test anywhere in the service mesh
- ▶ Discoverable tests using a /tests endpoint
- ▶ Monitor canary health in New Relic
- ▶ Promote to Canary Partial

Continuous Delivery Pipeline



- ▶ Receive partial production load
- ▶ Monitor canary health in New Relic
- ▶ Validate response codes
- ▶ Measure throughput
- ▶ Promote to general availability

Continuous Delivery Pipeline

Sentinel



Continuous Delivery Pipeline

Sentinel

Sentinel

Switch back to V1 UI

Search Services

Canaries

Search

Sort By

Most Recent

Severity

service

In Progress

lighthouse-task-service

In Progress

elastic-reporting-service

SUCCESS

lighthouse-task-service

In Progress

Phase

FULL

Last Result

REENQUEUE (a minute ago)

Rollback

Stop

Promote

task-service

SUCCESS

Phase

FULL

Last Result

DONE (a day ago)

google-location-service

SUCCESS

Phase

FULL

Last Result

DONE (a day ago)

offers-ui

SUCCESS

Phase

FULL

Last Result

DONE (a day ago)

Continuous Delivery Pipeline

Sentinel

Sentinel Switch back to V1 UI

Canaries

Search

service In Progress

Phase

Last Result None Yet

Rollback

Stop

Promote

offer-service SUCCESS

Phase

Last Result FULL DONE (an hour ago)

email-marketing-service SUCCESS

Phase

Last Result FULL DONE (6 hours ago)

task-service SUCCESS

Phase

Last Result FULL DONE (a day ago)

search

unsubscribe-service

http-thrift-service

contact-appointment-materialization

contact-search-fields-metadata-service

model-builder-service

website-layout-mapping-service

elastic-reporting-service

email-marketing-service

lighthouse-marketing

salesforce-transform-service

promotions-service

available-hours-service

contact-search-fields-service

contact-search-service

seoanatomy-service

business-info-service

communication-renderer-service

Search Services

Sort By Most Recent Severity

reporting-service SUCCESS

Phase

Last Result FULL DONE (36 minutes ago)

client-budget-tool SUCCESS

Phase

Last Result FULL DONE (6 hours ago)

yncer-ui SUCCESS

Phase

Last Result FULL DONE (a day ago)

-ui SUCCESS

Phase

Last Result FULL DONE (a day ago)

yodle®

Continuous Delivery Pipeline

Sentinel

cerebro

ID: cd5cbdbb-2fc2-4d11-95e8-f11fc0ea6d63

Status: Complete

Started: Last Friday at 12:19 PM

Canaries ▾

▼ Last Canary Check: SUCCESS

Last Friday at 12:26 PM

All service instances are still alive

Service is discoverable.

All service instances are still healthy

Triggering REENQUEUE because of canary test results

View All

▼ Last Test Result: CONTINUE

4 days ago

Test GET /canarytest/469232f0-615c-40fb-8428-30490aff149a

Host http://dev-mesos[REDACTED]dle.com:31173

View All

Continuous Delivery Pipeline

Sentinel

cerebro

ID: cd5cbdbb-2fc2-4d11-95e8-f11fc0ea6d63
Status: Complete
Started: Last Friday at 12:19 PM

☑ Last Canary Check: SUCCESS Last Friday at 12:26 PM

All service instances are still alive
Service is discoverable.
All service instances are still healthy
Triggering REENQUEUE because of canary test results

[View All](#)

Canaries ▾

- Last Friday at 12:19 PM - DONE - FULL
- 05/25/2016 - DONE - FULL
- 05/23/2016 - DONE - FULL
- 05/23/2016 - DONE - FULL
- 05/23/2016 - ROLLBACK - FULL
- 05/23/2016 - ROLLBACK - FULL
- 05/12/2016 - DONE - FULL
- 05/11/2016 - DONE - FULL
- 05/11/2016 - DONE - FULL
- 05/11/2016 - DONE - FULL
- 04/27/2016 - DONE - FULL
- 04/25/2016 - DONE - FULL
- 04/20/2016 - DONE - FULL
- 04/18/2016 - DONE - FULL
- 04/18/2016 - DONE - FULL
- 04/14/2016 - DONE - FULL
- 04/14/2016 - ROLLBACK - FULL

Canaries ▾

4 days ago

40fb-8428-30490aff149a
ev.yodle.com:31173

[View All](#)

Continuous Delivery Pipeline

Sentinel

cerebro

II

Status

Started

Last Canary

All service instan

Service is discov

All service instan

Triggering REEN

Canary Tests: cerebro

✔ GET /canarytest/469232f0-615c-40fb-8428-30490aff149a

Last Friday at 12:22 PM

✔ GET /canarytest/13037825-1b86-449c-bf8c-f1287124947a

Last Friday at 12:22 PM

✔ GET /api/v1/deployments/cerebro

Last Friday at 12:22 PM

✔ GET /api/v1/deployments/cerebro

Last Friday at 12:22 PM

✔ GET /api/v1/deployments/iamanappthatdoesnotexist

Last Friday at 12:22 PM

✔ GET /api/v1/deployments/iamanappthatdoesnotexist

Last Friday at 12:22 PM

Canaries ▾

4 days ago

490aff149a

81173

View All

Continuous Delivery Pipeline

Canary Checks: cerebro

- ✔ #7: Canary Monitoring Completed
- 🔄 #6: No Fatal Errors Found For Canary. Will Continue Monitoring
- 🔄 #5: No Fatal Errors Found For Canary. Will Continue Monitoring
- All service instances are still alive
Service is discoverable.
All service instances are still healthy
Triggering REENQUEUE because of canary test results
- 🔄 #4: No Fatal Errors Found For Canary. Will Continue Monitoring
- 🔄 #3: No Fatal Errors Found For Canary. Will Continue Monitoring
- 🔄 #2: No Fatal Errors Found For Canary. Will Continue Monitoring
- 🔄 #1: No Fatal Errors Found For Canary. Will Continue Monitoring

Canaries ▾

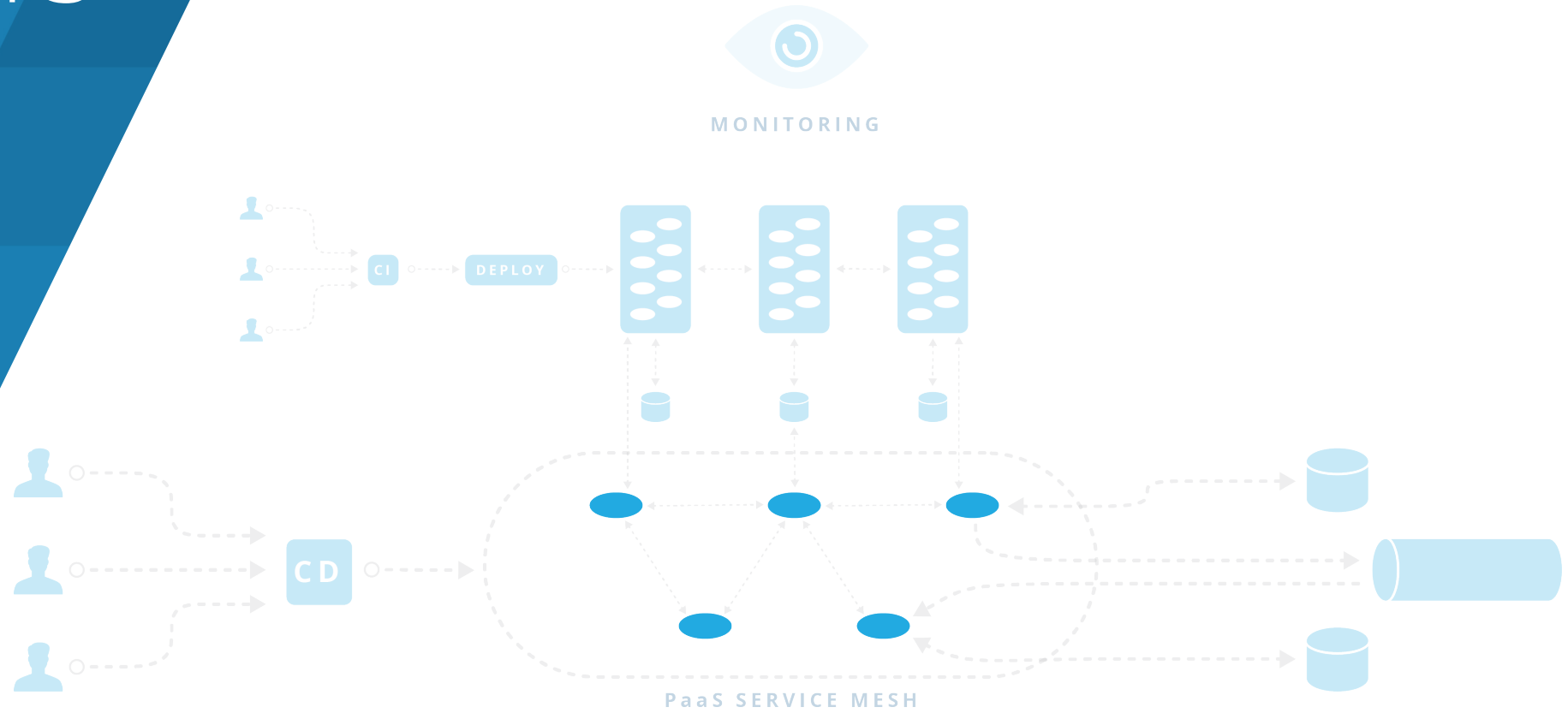
4 days ago

490aff149a
1173

[View All](#)

ances: 2

Containers



Containers Bring Simplicity

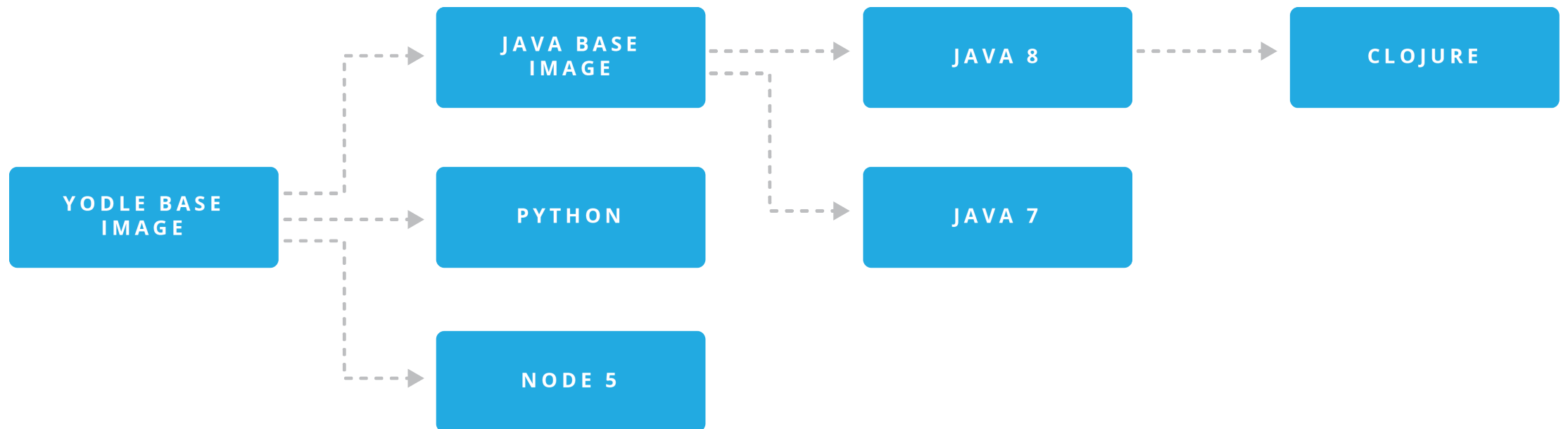
Standardization is required

- ▶ Polyglot environments buck standardization
- ▶ Micro-service environments increase complexity
- ▶ Operational complexity can grow unbounded
- ▶ Developers own the runtime
- ▶ Common runtime from an operator's standpoint
- ▶ Tooling provides consistent deployments

Containers Bring Simplicity

Hierarchical Container Images

- ▶ How do you roll out environmental changes when you have 200 different container builds?



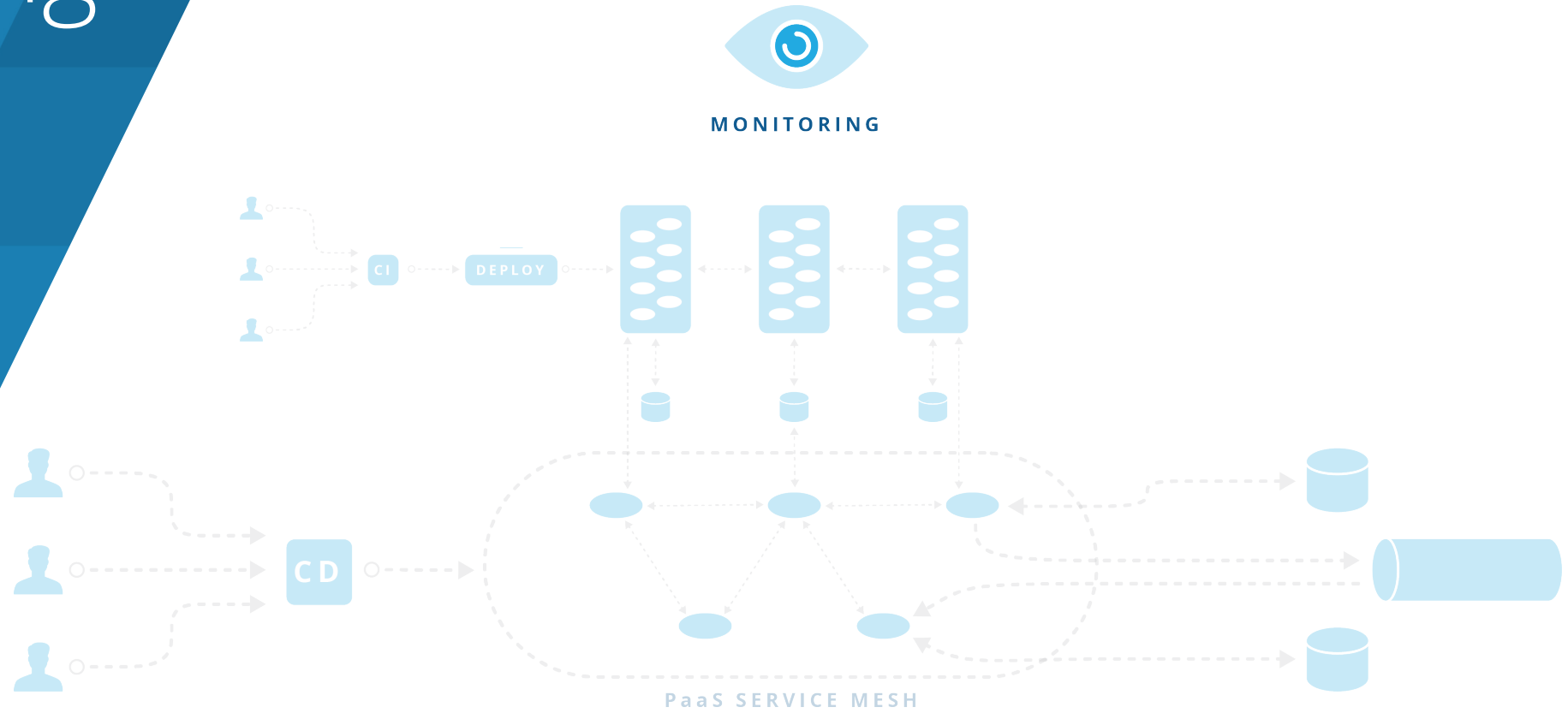
Containers Bring Simplicity

Containers make a mess

- ▶ Docker host machines were littered
- ▶ Docker registry is littered with old images
- ▶ Developed a tagging process



Monitoring



Increased Complexity Increased Requirements

Legacy Monitoring not cutting it

- ▶ Designed for testing and monitoring infrastructure
- ▶ Needed application performance management
- ▶ Wanted something that would scale with us with little effort

Increased Complexity Increased Requirements

Graphite and Grafana

- ▶ Dropwizard metrics to report data
- ▶ Teams built custom dashboards
- ▶ Too much manual effort
- ▶ No alerting

Increased Complexity Increased Requirements

Enter the Hackathon

- ▶ New Relic Monitoring For Microservices
- ▶ Simple – just add an agent
- ▶ Detailed per application dashboards out of the box
- ▶ Single score to focus attention (Useful for initial canary implementation)
- ▶ Basic alerting

Increased Complexity Increased Requirements

100 Apps in 100 Days

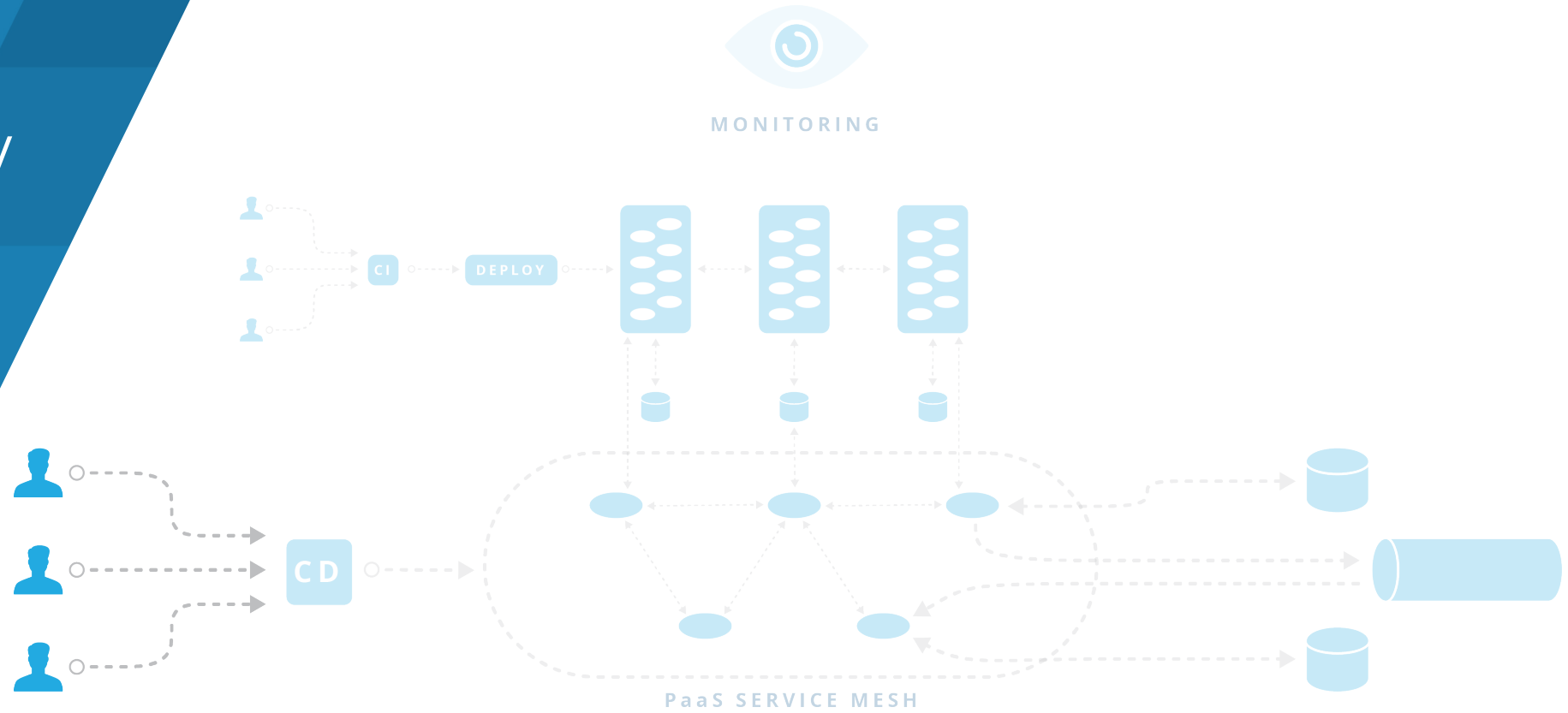
- ▶ Made use of our base containers
- ▶ Rolled out monitoring to every application in the fleet
- ▶ Suddenly we had visibility everywhere.
- ▶ Some Limitations
 - No good docker support (this is better now)
 - Services graphs aren't dynamically generated

Increased Complexity Increased Requirements

Finding root causes

- ▶ Hundreds of Dashboards
- ▶ Hundreds of Individual Service Nodes
- ▶ Finding root causes in complex service graphs is difficult
- ▶ Anomalies from individual service nodes difficult to detect
- ▶ Still looking for a good solution

Source Repository Complexity



Source Repository Complexity

Source Code Management

- ▶ Organizational scheme to help think about it
- ▶ Hound to help with code searching
- ▶ Repo tool to help keep up-to-date
- ▶ Upgrading libraries is a challenge

Source Repository Complexity

Dependency Management

Components

Displaying 1 to 10 of 801

vantage

View most recent version: 7e5362e5b8064affc68e01282ad830de81ca28fd

org.eclipse.jetty:jetty-jsp

View most recent version: 9.2.14.v20151106

org.glassfish.web:javax.servlet.jsp

View most recent version: 2.3.2

org.eclipse.jetty.orbit:javax.servlet.jsp.jstl

View most recent version: 1.2.0.v201105211821

org.glassfish.web:javax.servlet.jsp.jstl

View most recent version: 1.2.2

javax.servlet:javax.servlet-api

View most recent version: 3.1.0

javax.servlet.jsp:javax.servlet.jsp-api

View most recent version: 2.3.1

Source Repository Complexity

Dependency Management

All Components / sync-service

sync-service

Versions

0c7cfb497fcba12745486533979a0630250b2538



4e4772be1461bcd1fb9a15fb1f547f5fe9db9b28



a5e4c833f4c0590e515355f001230ca91e760e1e



da140031435014a8903c3d7008f0efffd470b8d3



Source Repository Complexity

Dependency Management

Issues

There are currently no known issues with sync-service:0c7cfb497fcb12745486533979a0630250b2538.

Depends On

Displaying 1 to 10 of 280

<>

<div>com.yodle:admin:1.4</div> <div>View all versions of com.yodle:admin</div>	<div>!</div>
<div>com.yodle:sabre-client:1.4.5-29</div> <div>View all versions of com.yodle:sabre-client</div>	<div>!</div>
<div>aopalliance:aopalliance:1.0</div> <div>View all versions of aopalliance:aopalliance</div>	<div>!</div>

Source Repository Complexity

Dependency Management

Issues

YO-22442 - MINOR: The client side timed lock can inadvertently leak ephemeral nodes if requestCompletion is called on an application already in the process of executing that job.

Dependents

[user-service:0c190ca9d1bb7b9a51e7358913785e8c8e2c8cbb](#)

[View all versions of user-service](#)

[sync-service:0c7cfb497fcba12745486533979a0630250b2538](#)

[View all versions of sync-service](#)

Depends On

Warning: These dependencies have been implicitly reported by other components that depend on this component. They only represent the dependencies directly requested by this component as we cannot determine how this component's build script would resolve transitive dependencies.

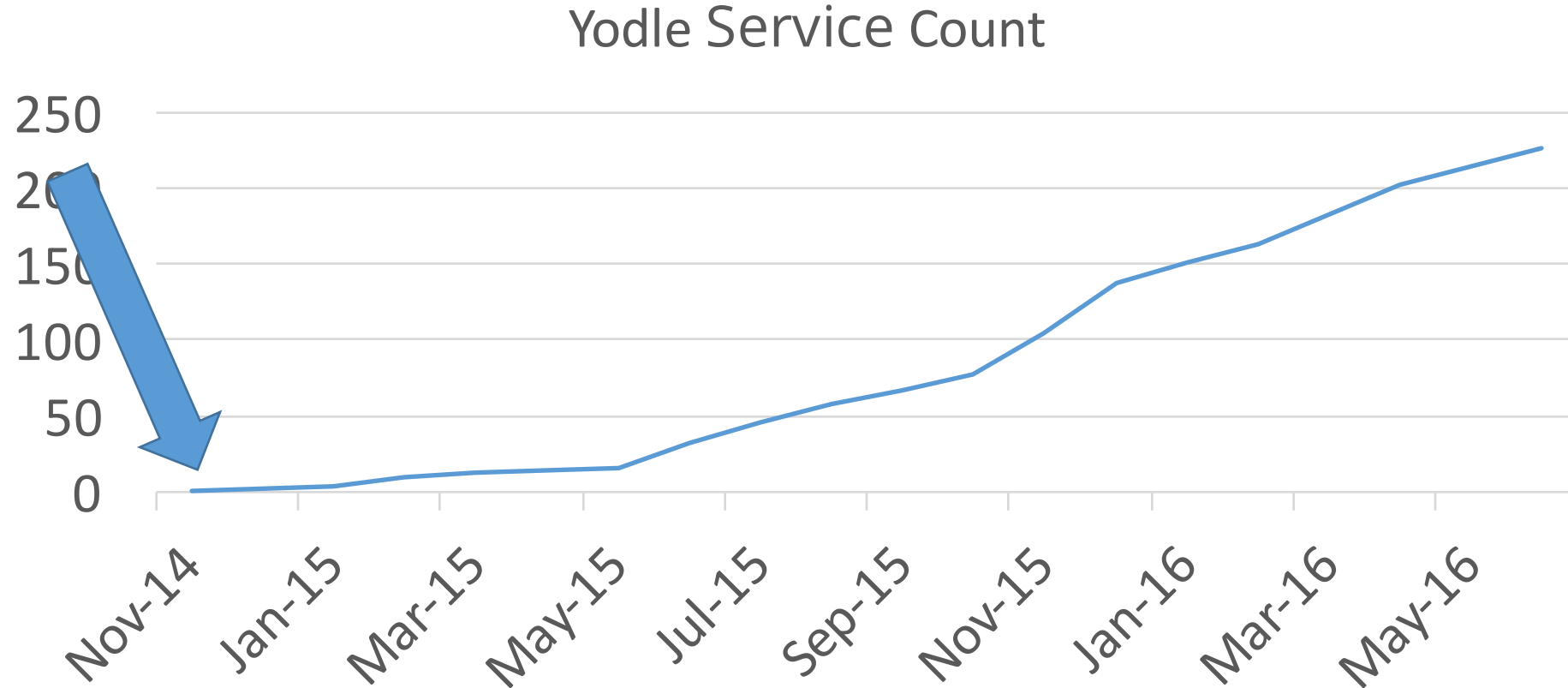
Source Repository Complexity

Build tooling

- ▶ Many build systems don't directly allow scripting
- ▶ Bamboo definitely doesn't
- ▶ Build tooling iterations are painful
- ▶ Managing Bamboo build and deploy plans at scale is hard

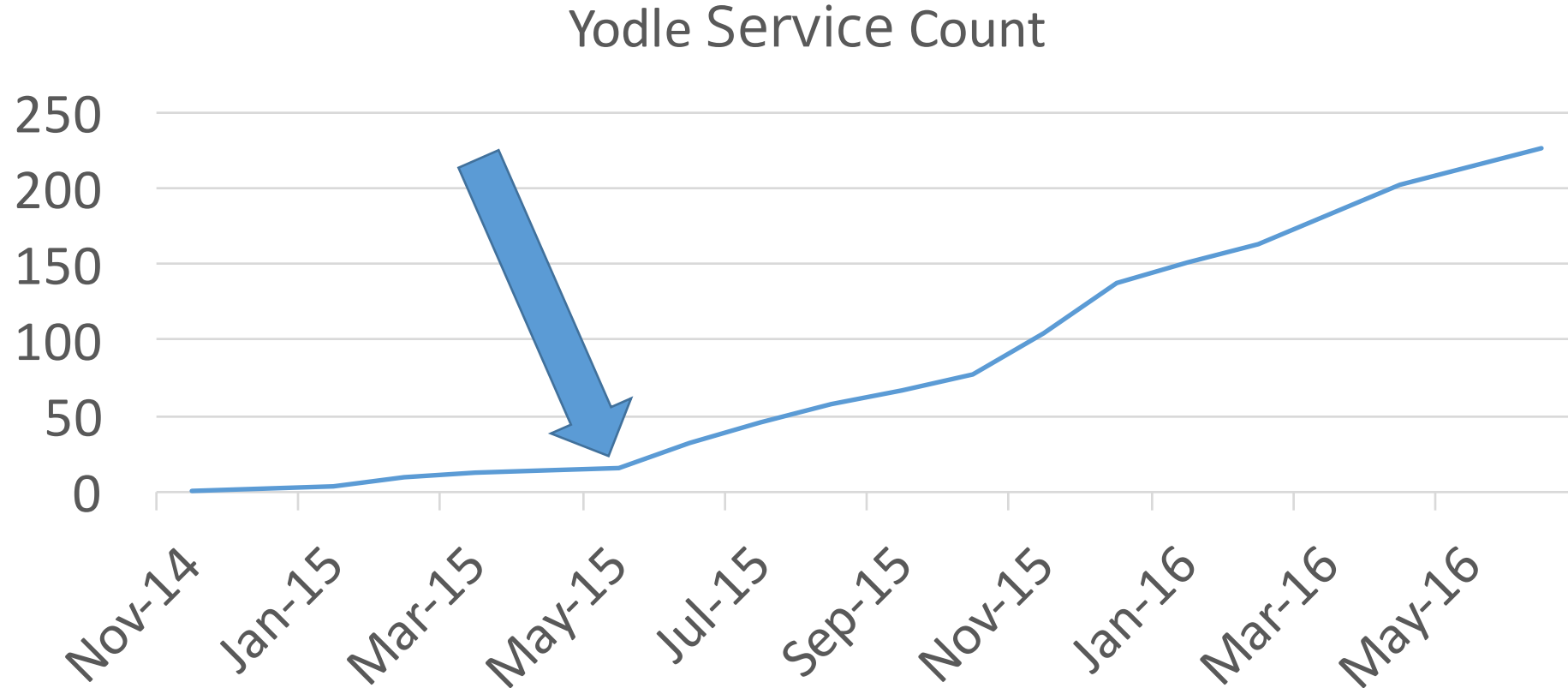
Build and Deploy Tooling

Existing Build Tooling



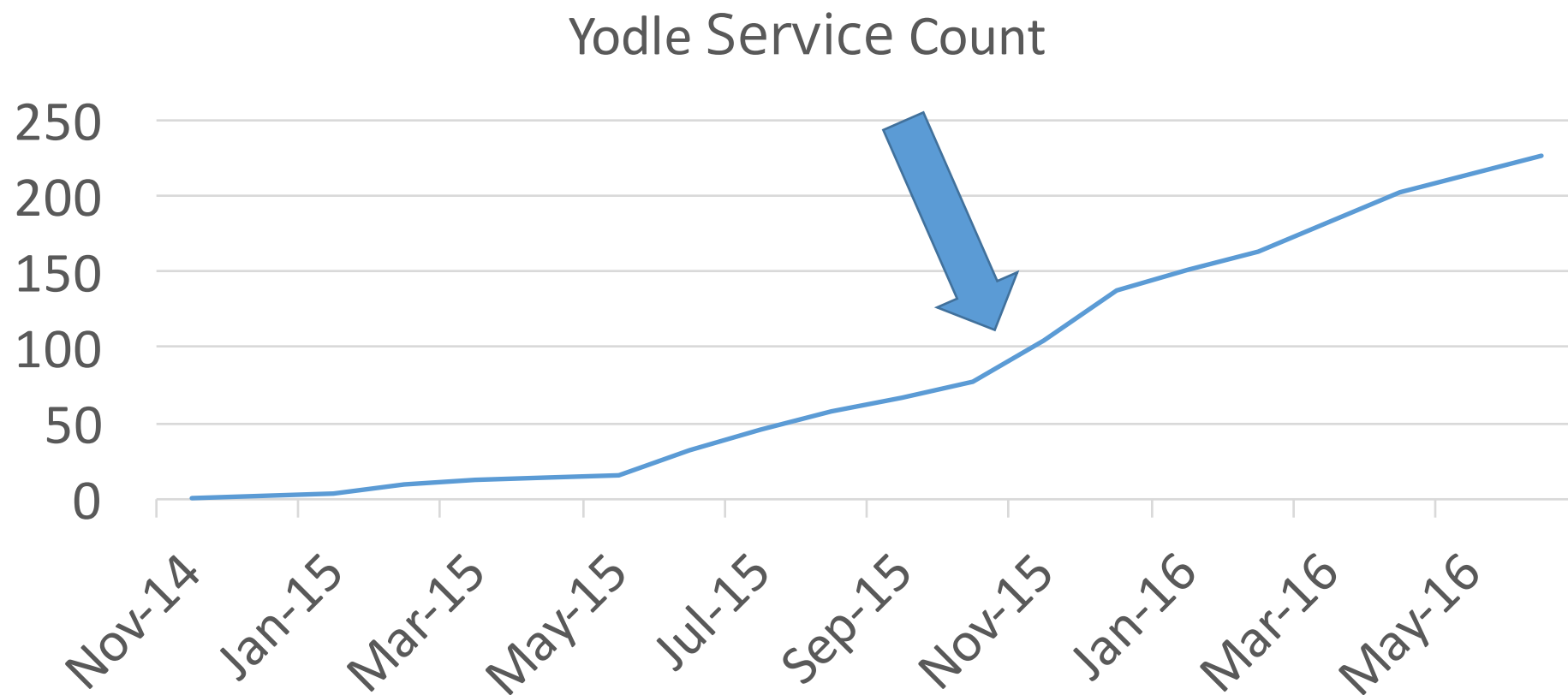
Build and Deploy Tooling

Directly to Marathon configurations in Bamboo



Build and Deploy Tooling

LaunchPad as an Abstraction Layer



Build and Deploy Tooling



Source Repository Complexity

Sentinel For Human Service Discovery



Source Repository Complexity

Sentinel For Human Service Discovery

Deployments

Yesterday at 5:51 PM
Phase: FULL

ID: launchpad/03098114-9ea4-442f-8c48-dad04b2ed5ac

Memory: 400

CPU: 0.15

Instances: 2

Scale

Restart

Suspend

Undeploy

Service Logs

Host	Start Date	Status	Health	Container Logs
dev-mesos-██████████yodle.com:31860	Yesterday at 5:51 PM	Started	Healthy	Logs
dev-mesos-██████████yodle.com:31237	Yesterday at 5:51 PM	Started	Healthy	Logs

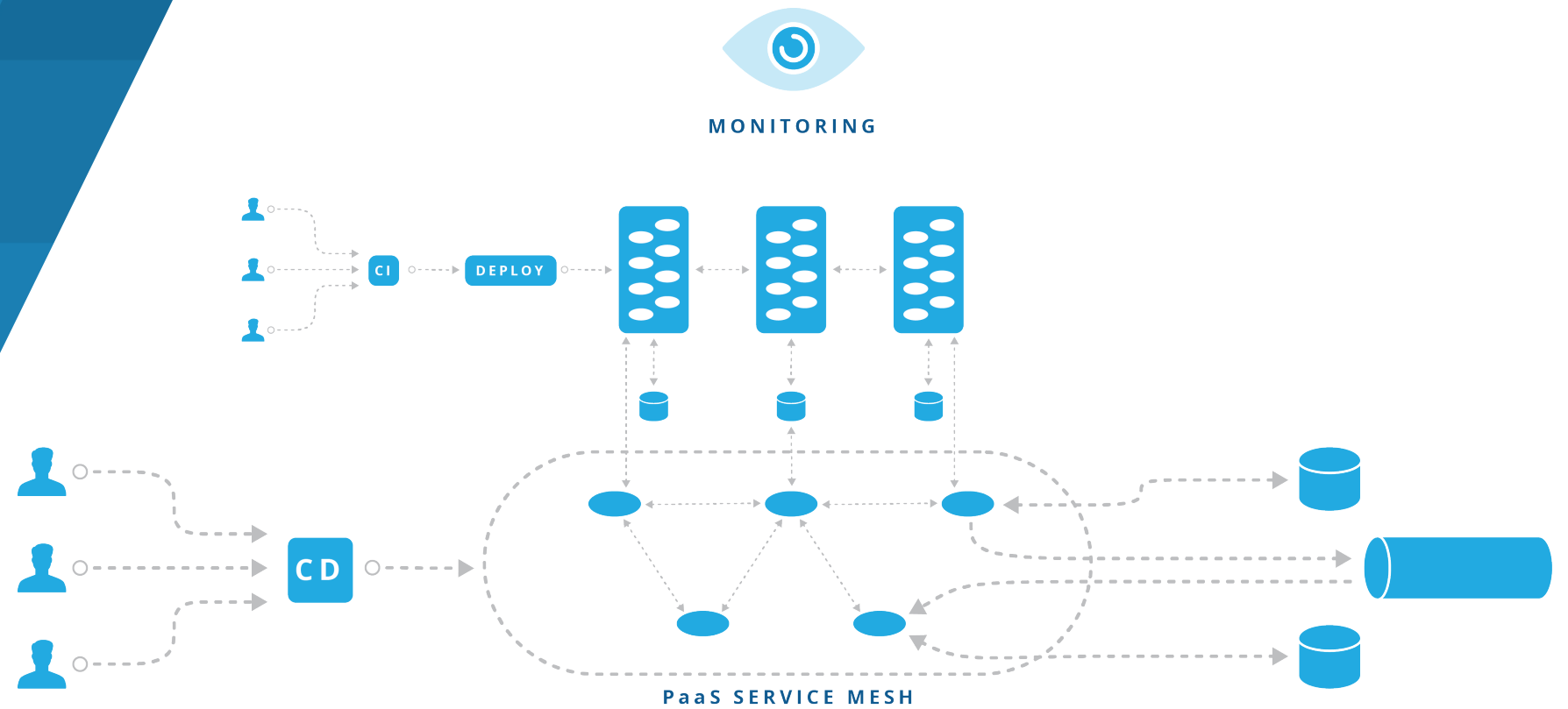
Source Repository Complexity

Sentinel For Human Service Discovery

Service Specific Properties

Project Category	Project Deployment	Project Description
Deployment Tools	Marathon	A tool for managing and monitoring deployments...
Project Language	Project Link	Project Name
Java	https://[REDACTED] ..	Cerebro
Project Owner	Project Repo	Project Revision
[REDACTED]@yodle.com	https://[REDACTED] ...	9f6be6c78d62366d40a3a07bf9f84e3833f5bd6d
Project Serviceprotocol	Service Containerid	Service Deploymentid
http	df3506f15cf4	e0196722-5f9e-458f-8cf4-6bf9e23cf2f4
Service Processname		
cerebro		

Conclusion



Plan for Challenges

Even if you aren't on the bleeding edge ...

- ▶ Every environment is different
- ▶ Legacy Applications present unique challenges
- ▶ Different business requirements
- ▶ Different trade-offs

Improved Agility

Every Hurdle Was Worth It Monthly Deployments





yodle[®]

We make it easy to grow and manage
profitable customer relationships

It's success simplified!