

Why Microservices Fail Steve Ross-Talbot - CTO, Estafet









Enterprise Integration

- Integration Experts for 15 years
- Specialists in Red Hat Middleware
- OpenShift 3

Agile Delivery

- Experts with Distributed Agile
- Estafet UK & Bulgaria (to scale delivery)





Mobile Development

Big Data & IoT

- Extending Integration Projects
- Apple iOS experts

- loT Big Data Delivery
- BI Reporting
- Delivering Large IoT Applications











- Co-author of the SOA Manifesto
 - Serial entrepreneur
- Professor of Distributed Computing

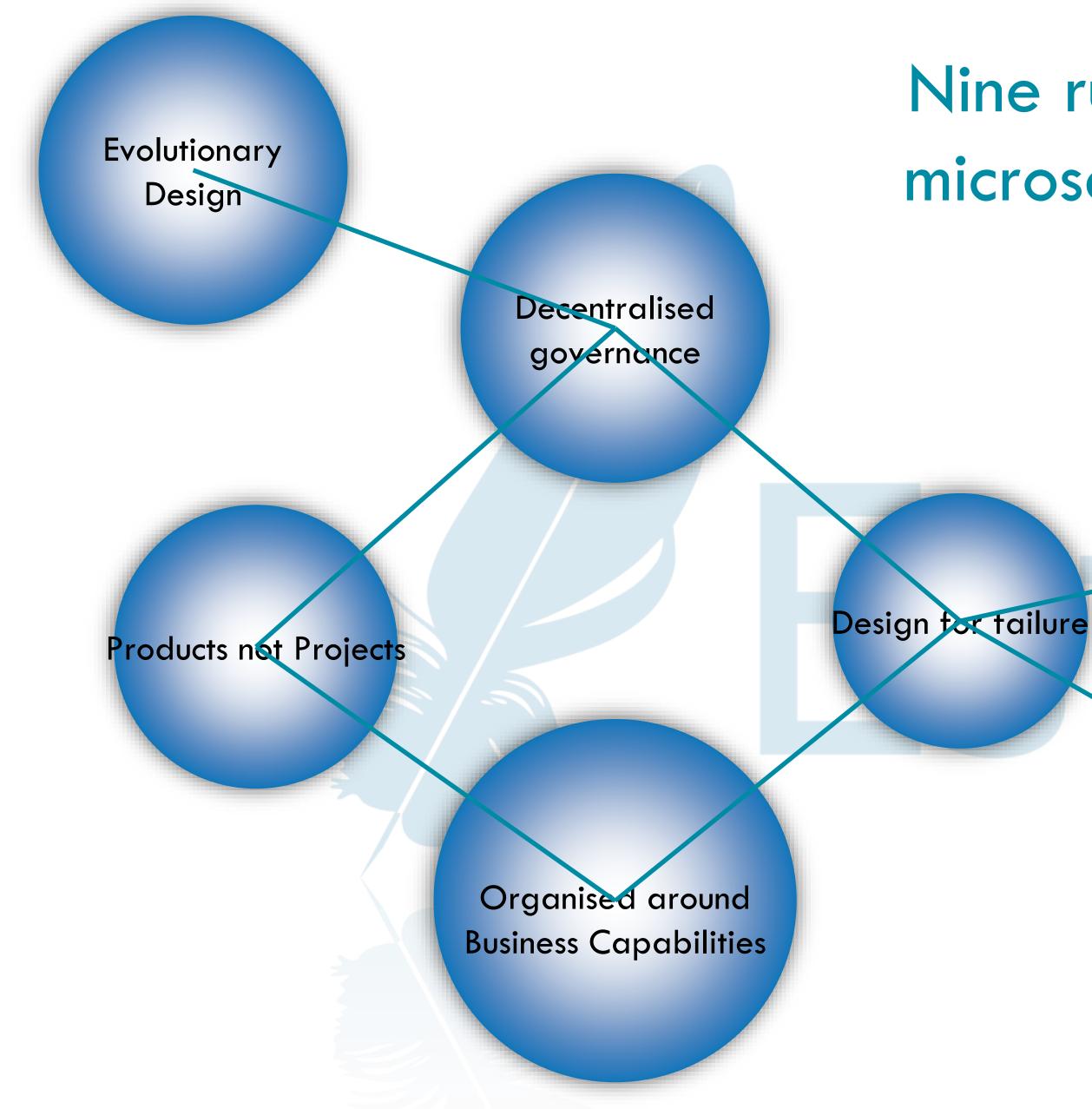


About Me

- Chief Technology Officer at Estafet
- Former chair of Web Services at W3C
- Former chair of W3C's Web Services Choreography WG







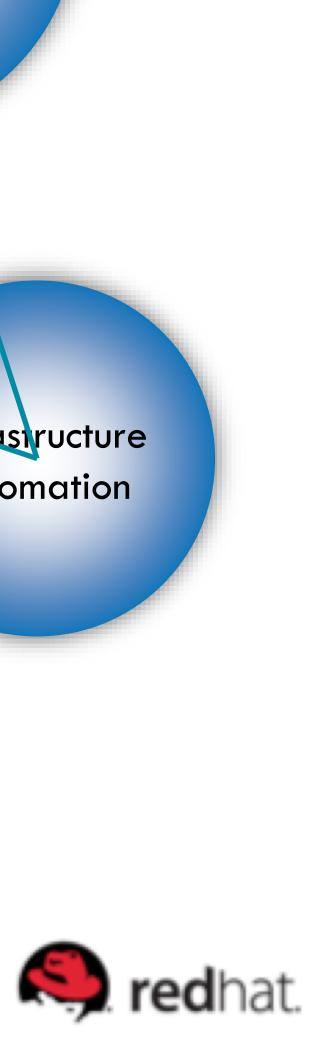
RED HAT FORUM Europe, Middle East & Africa

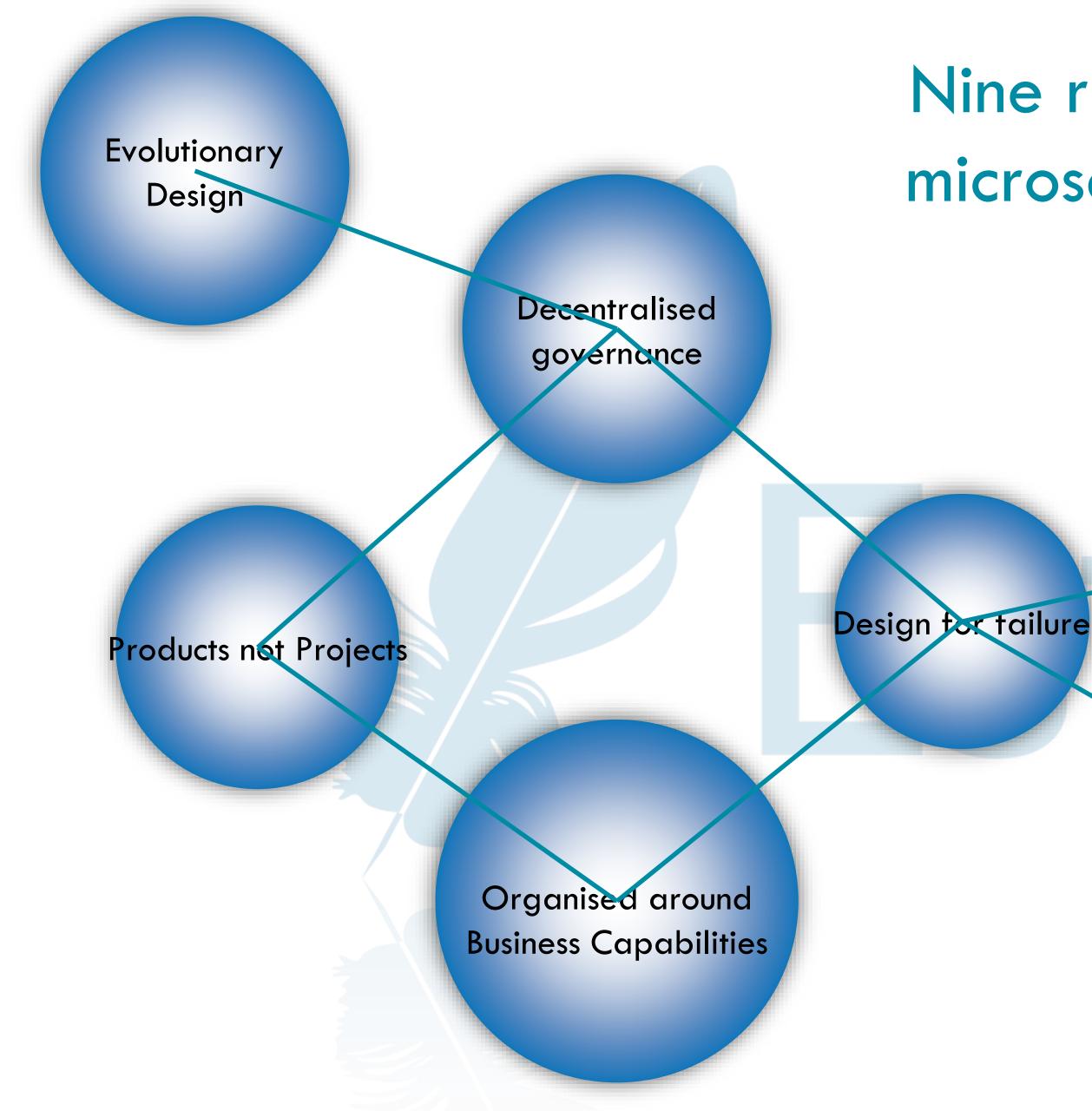
Nine rules of microservices

Componentisation via services

Smart endpoints and dumb pipes

Infrastructure automation





RED HAT FORUM Europe, Middle East & Africa

Nine rules of microservices

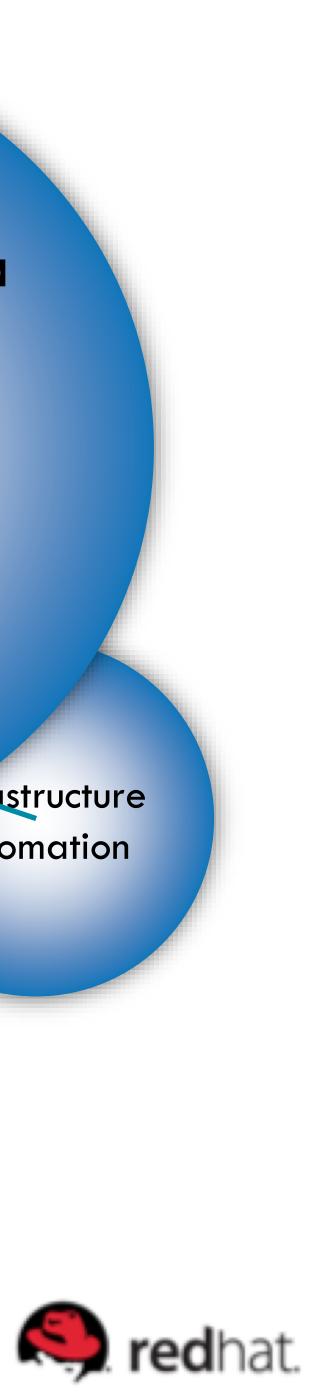
Sn

Gild

Componentisation via services

Independent deployability over reuse

mastructure automation





Decentralised governance

Products not Projects

Organised around **Business Capabilities**

Business value-led



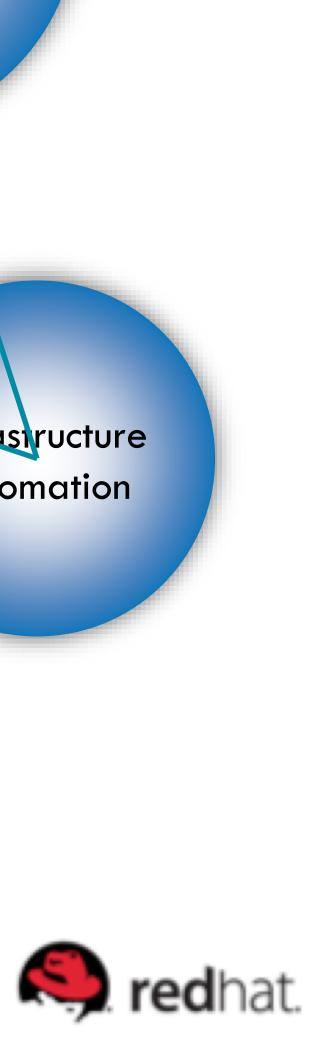
Nine rules of microservices

ailure

Componentisation via services

Smart endpoints and dumb pipes

Infrastructure automation





Decentralised governance

Products not Projects

workstreams

ed around ss Capabilities



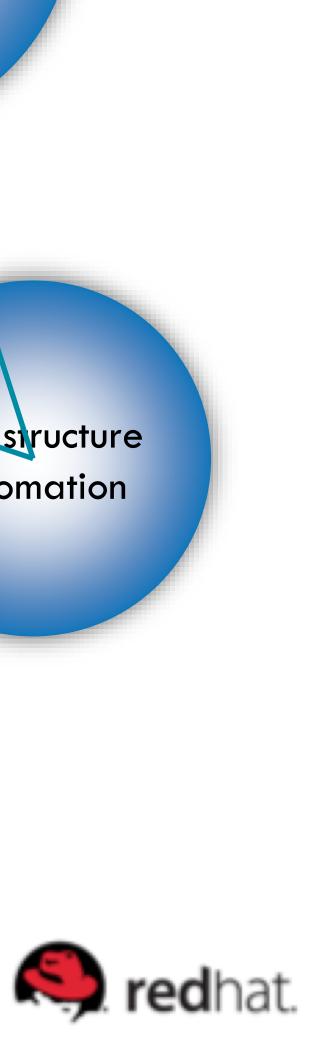
Design for tailure

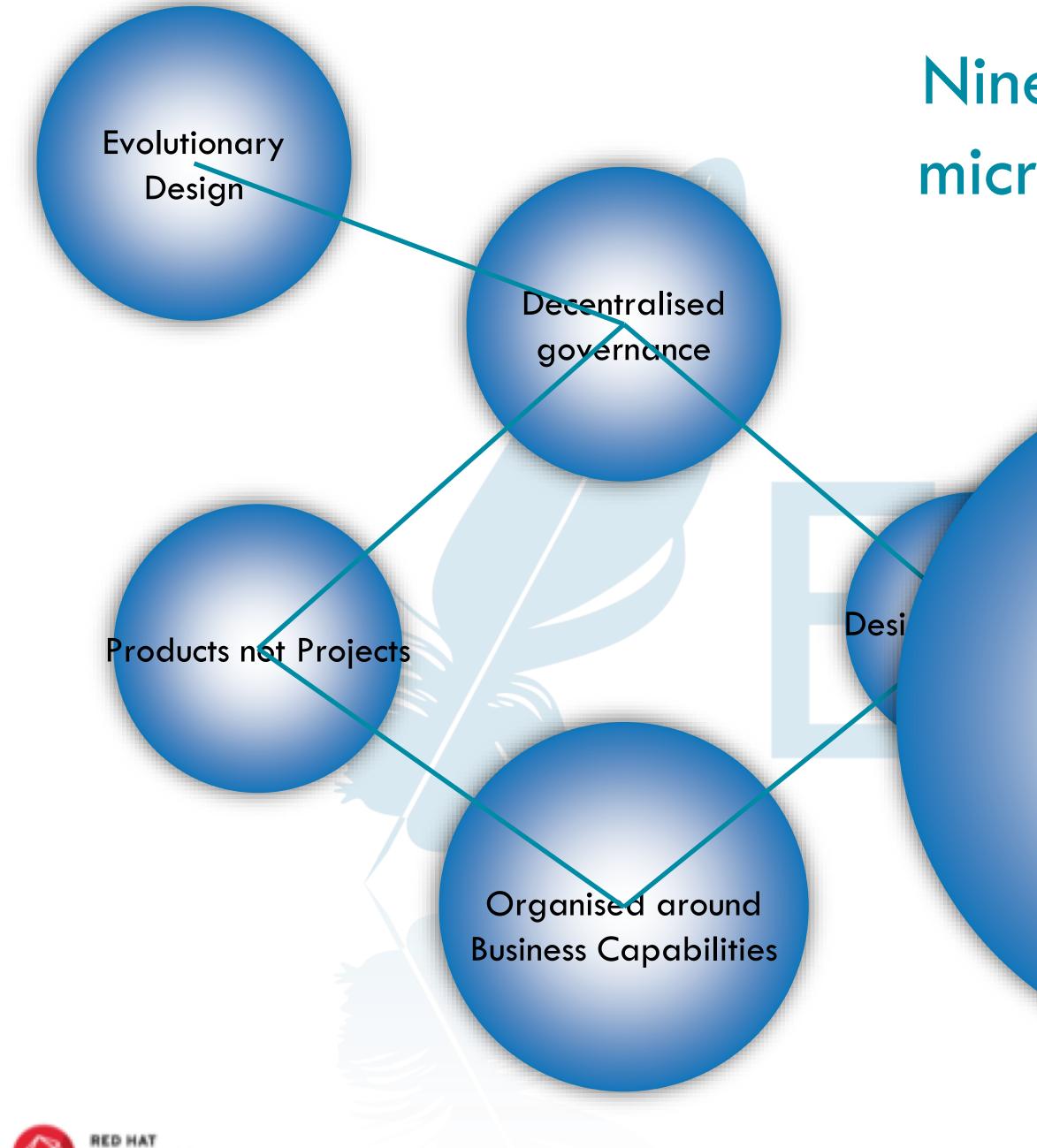
Nine rules of microservices

Componentisation via services

Smart endpoints and dumb pipes

Infrastructure automation





RED HAT FORUM Europe, Middle East & Africa

Nine rules of microservices

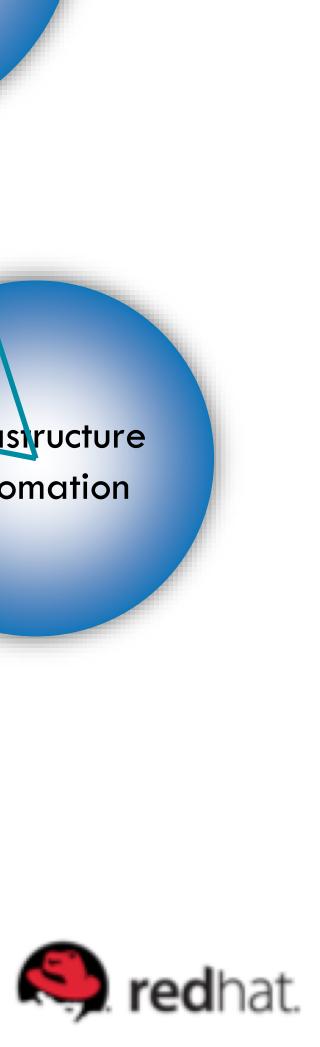
Componentisation via services

Smart endpoints and dumb pipes

Choreograph at the end points

ed data gement

Infrastructure automation



Evolutionary Design

Decentralised governance

Cross-functional teams make decisions

Design for tailure

Products not Projects

Organised around **Business Capabilities**

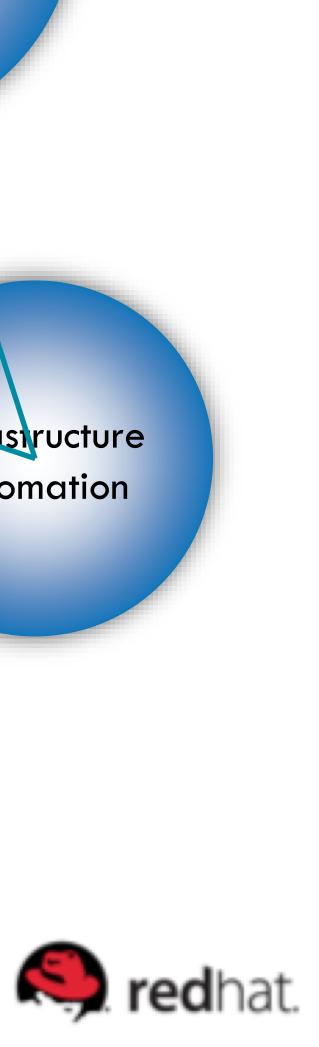


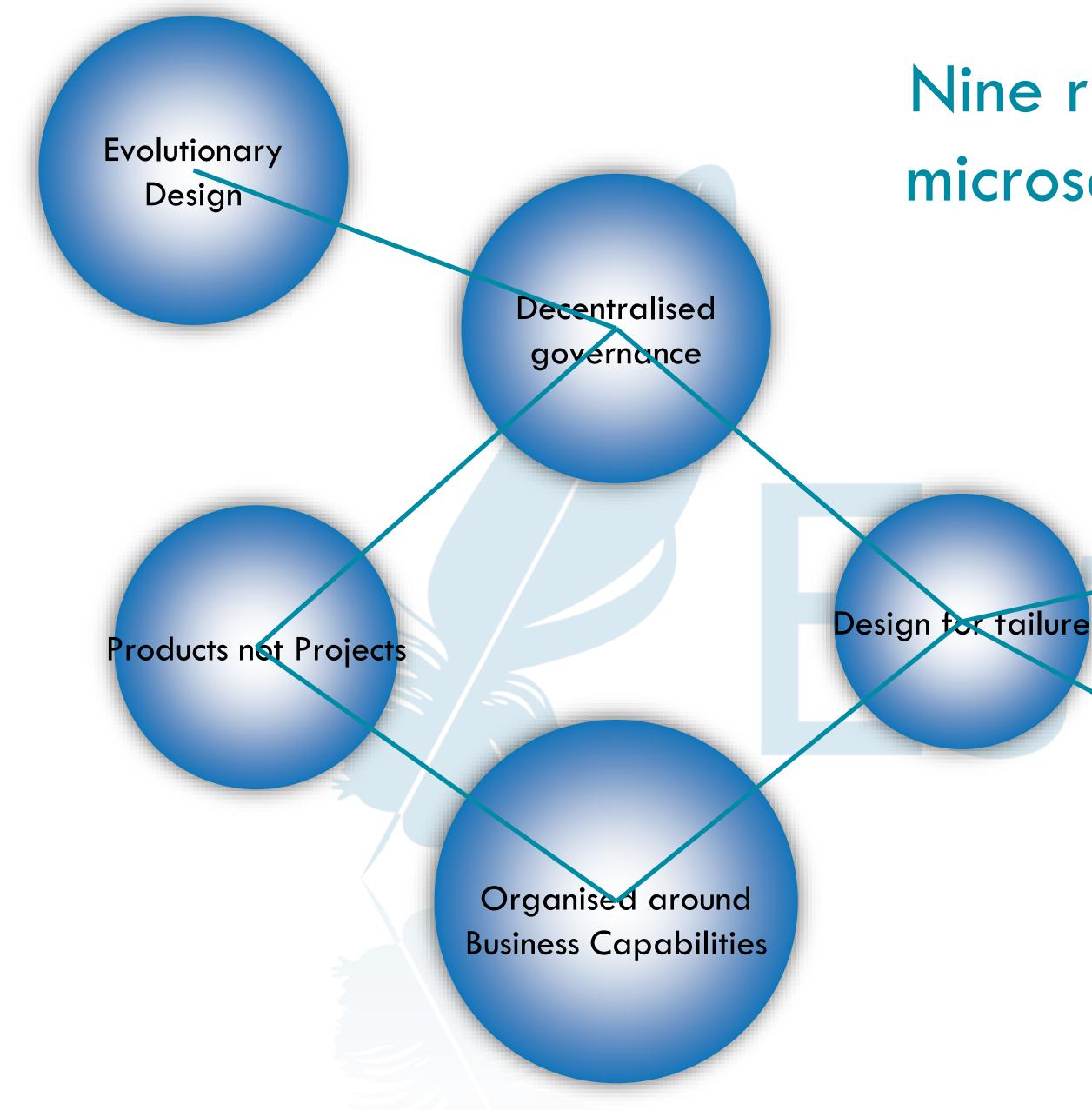
Nine rules of microservices

Componentisation via services

Smart endpoints and dumb pipes

Infrastructure automation





RED HAT FORUM Europe, Middle East & Africa

Nine rules of microservices

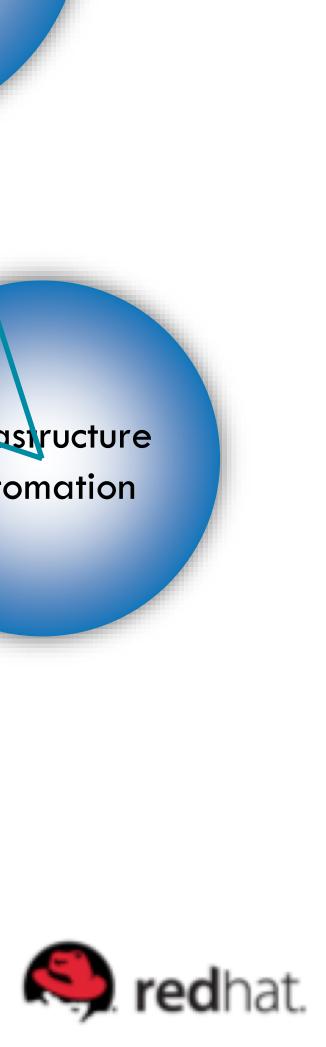
Componentisation via services

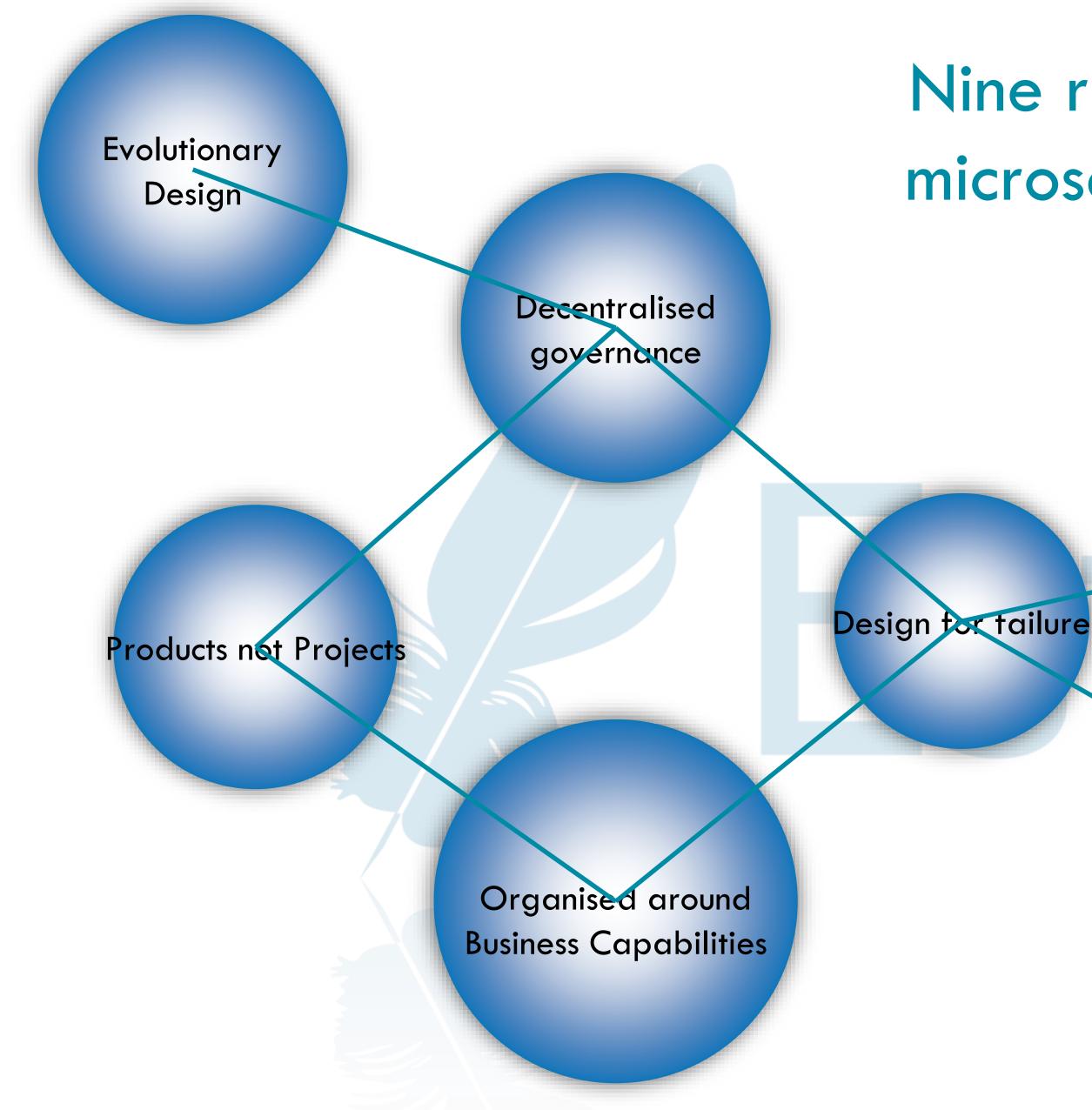
Smart endpoints and dumb pipes

Decentralised data management

Each team owns its data

Infrastructure automation





TORUM

Nine rules of microservices

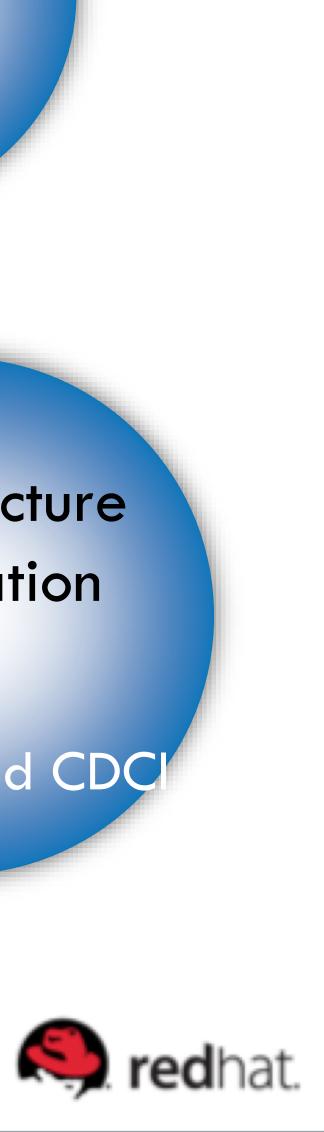
Componentisation via services

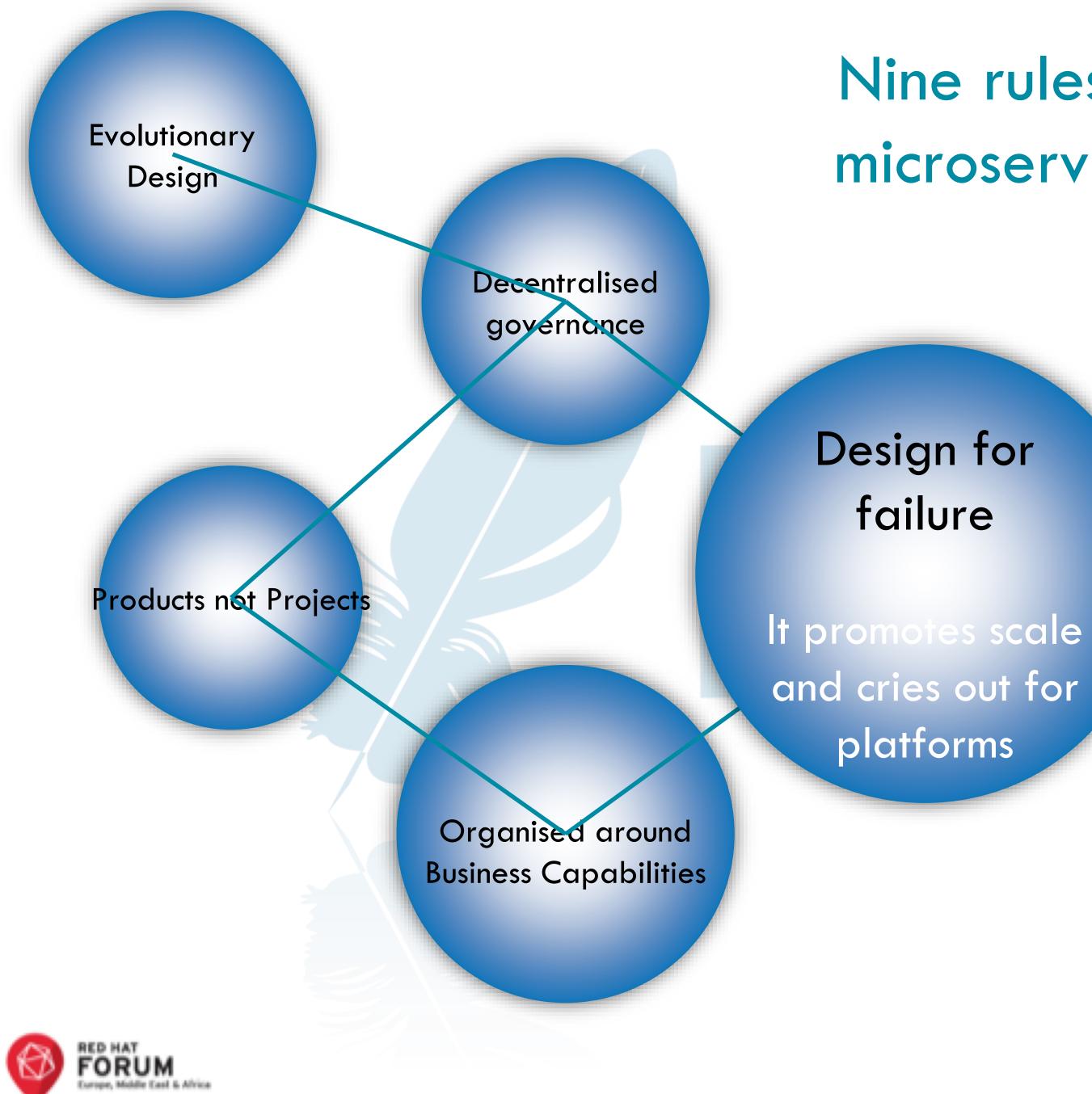
Smart endpoints and dumb pipes

Infrastructure automation

Decentralised data management

DevOps and CDC





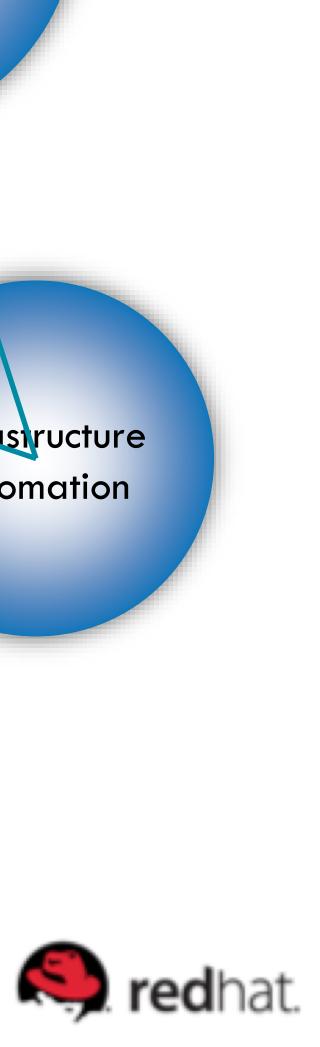
Nine rules of microservices

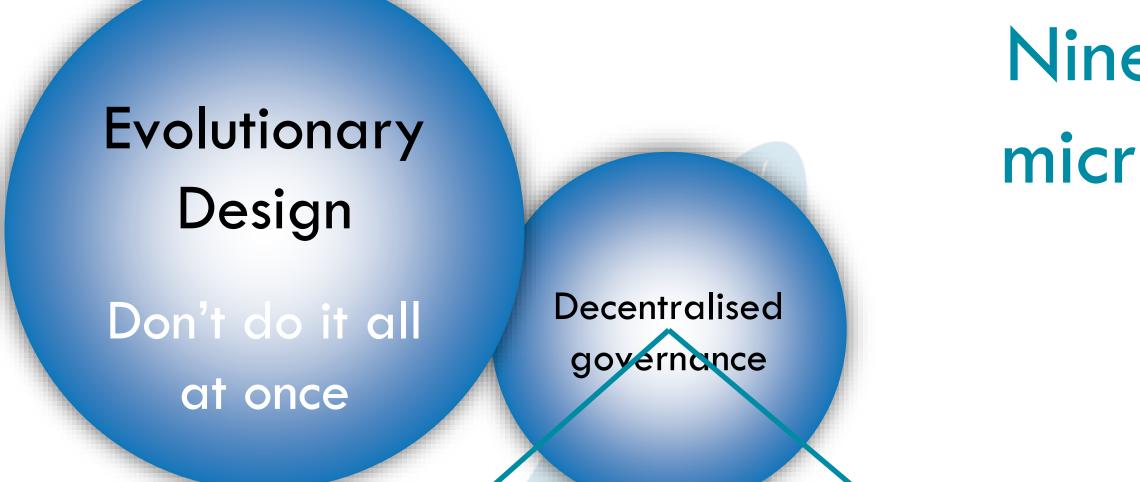
Componentisation via services

Smart endpoints and dumb pipes

Decentralised data management

Infrastructure automation





Products not Projects

Design for tailure

Organised around **Business Capabilities**

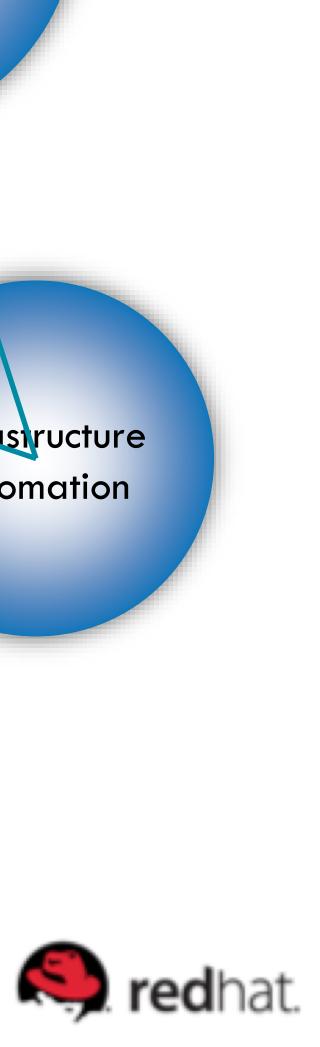


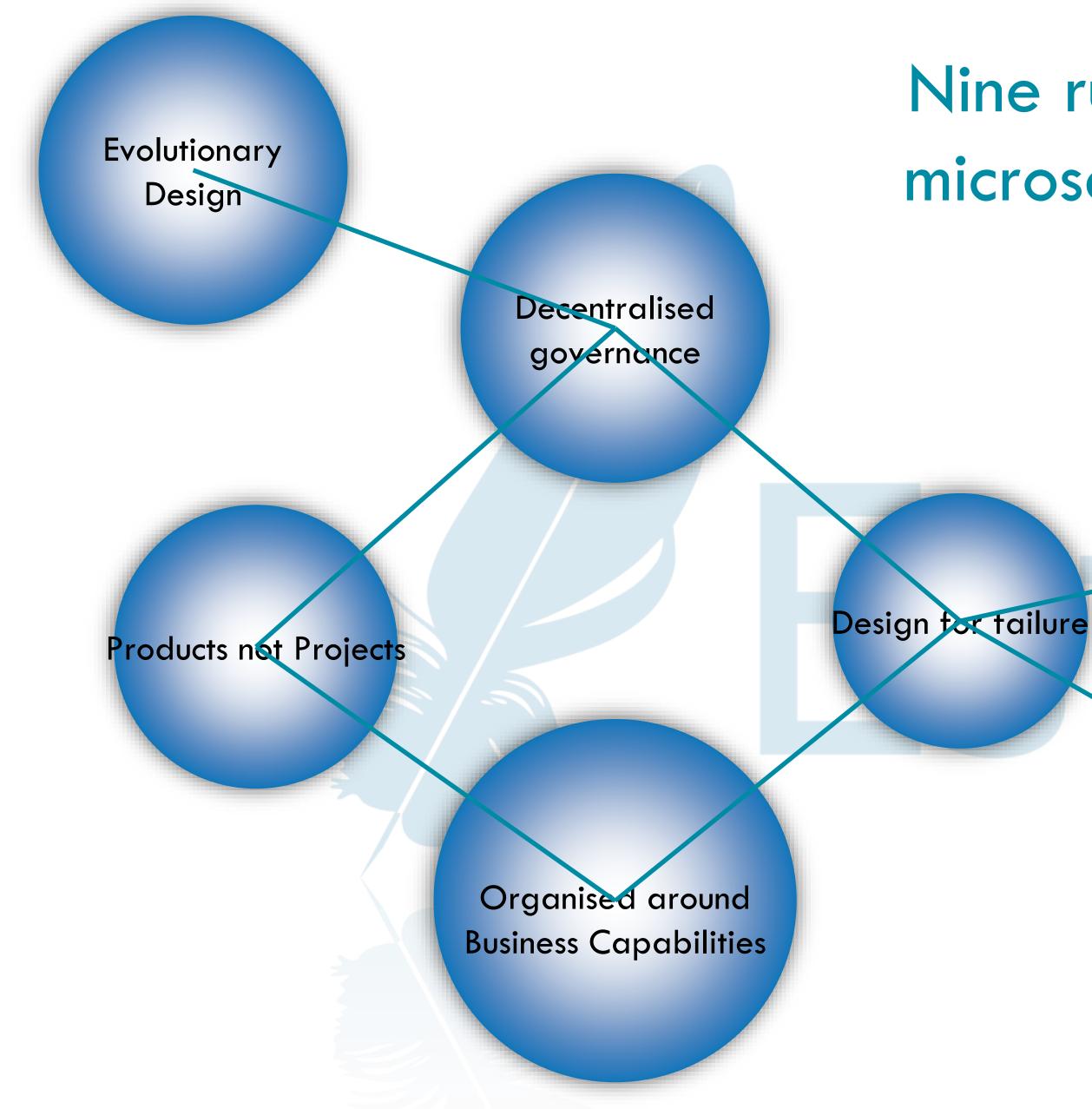
Nine rules of microservices

Componentisation via services

Smart endpoints and dumb pipes

Infrastructure automation





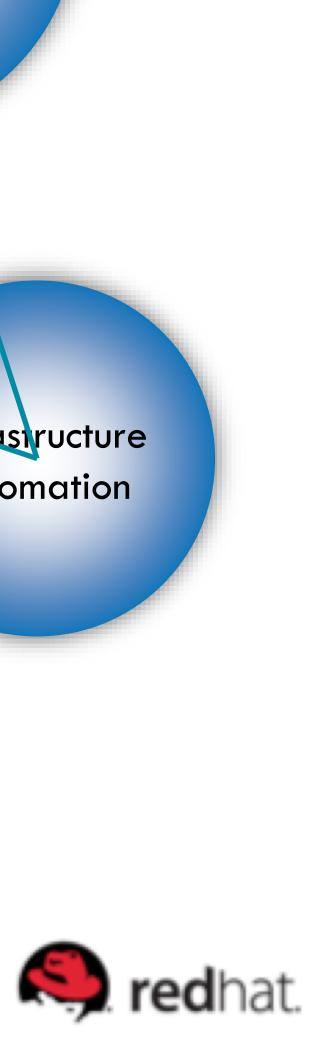
RED HAT FORUM Europe, Middle East & Africa

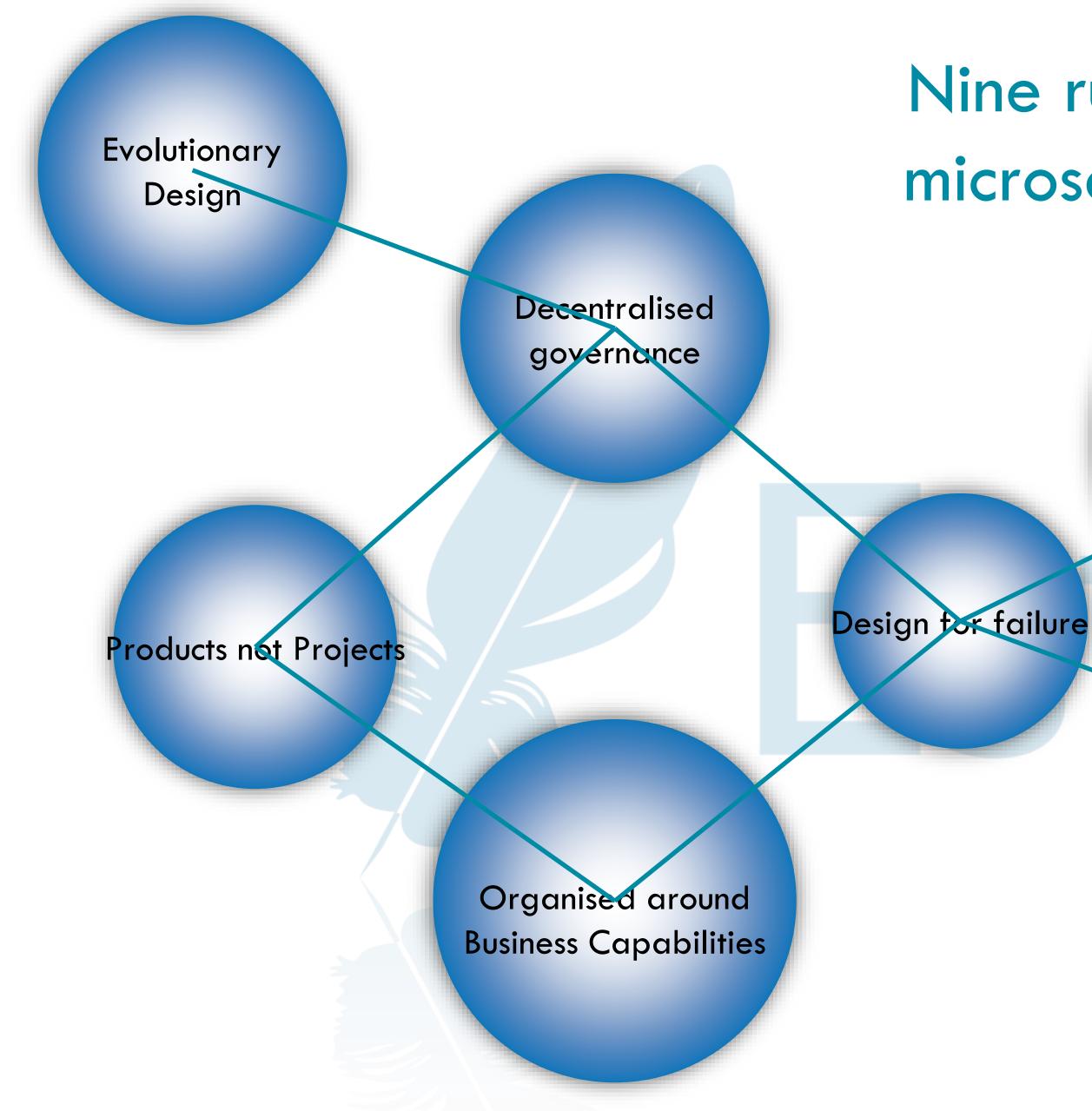
Nine rules of microservices

Componentisation via services

Smart endpoints and dumb pipes

Infrastructure automation







Nine rules of microservices

Componentisation via services

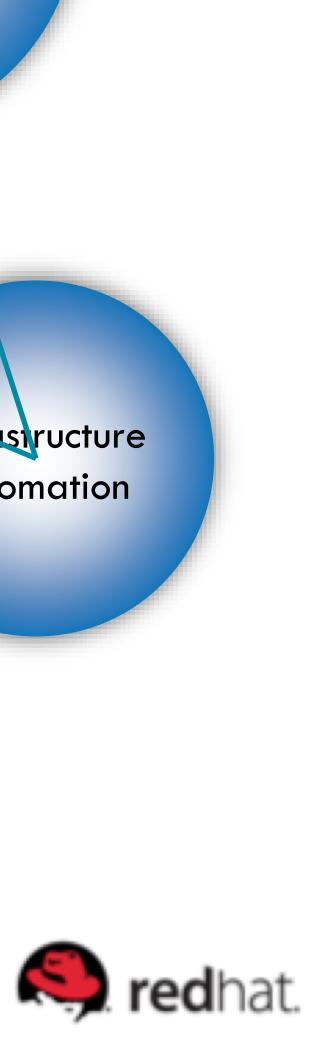
Smart endpoints and dumb pipes

DO NOT choreograph at endpoints

Decentralised data management

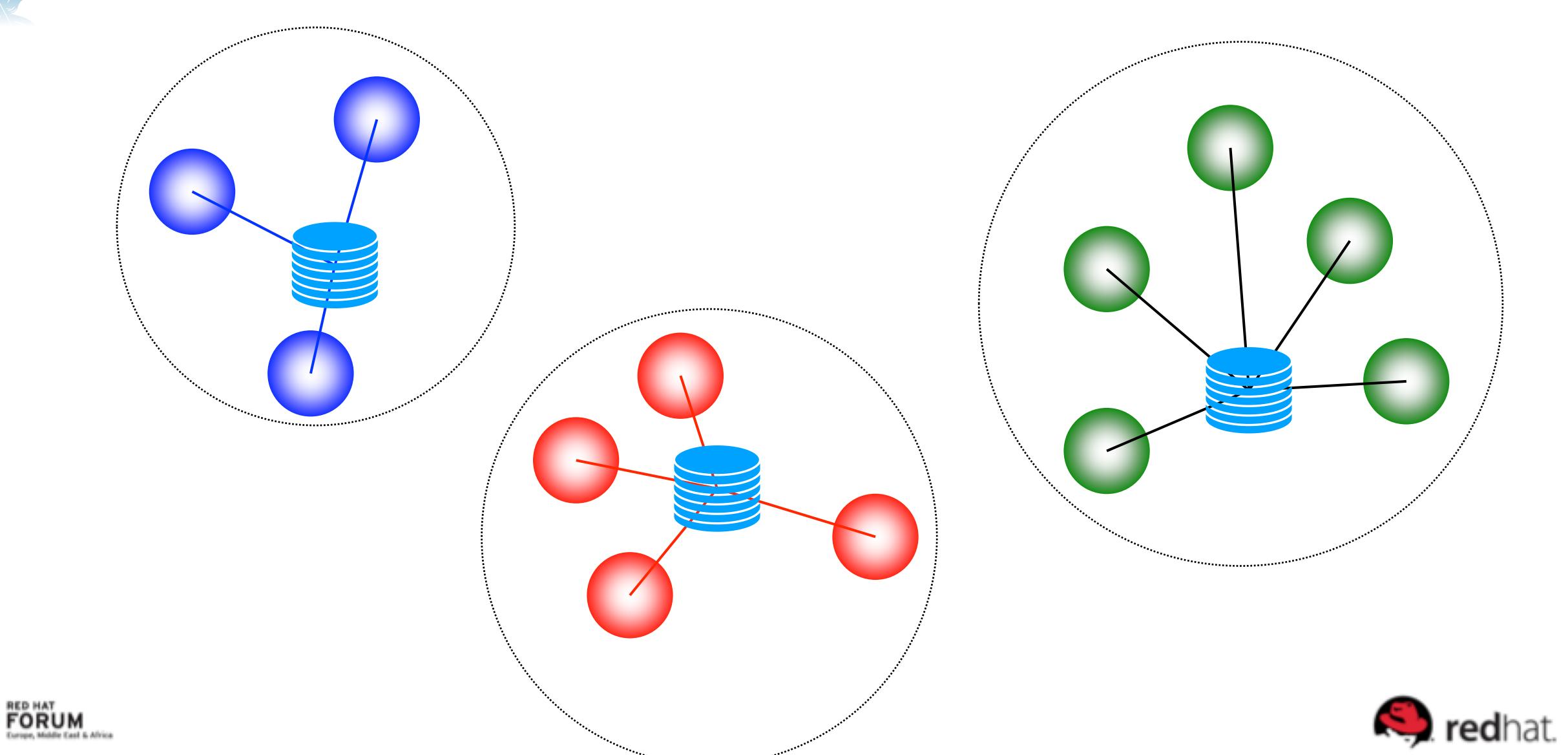
Each team owns its data: CONSISTENCY AS PROTOCOL

Infrastructure automation



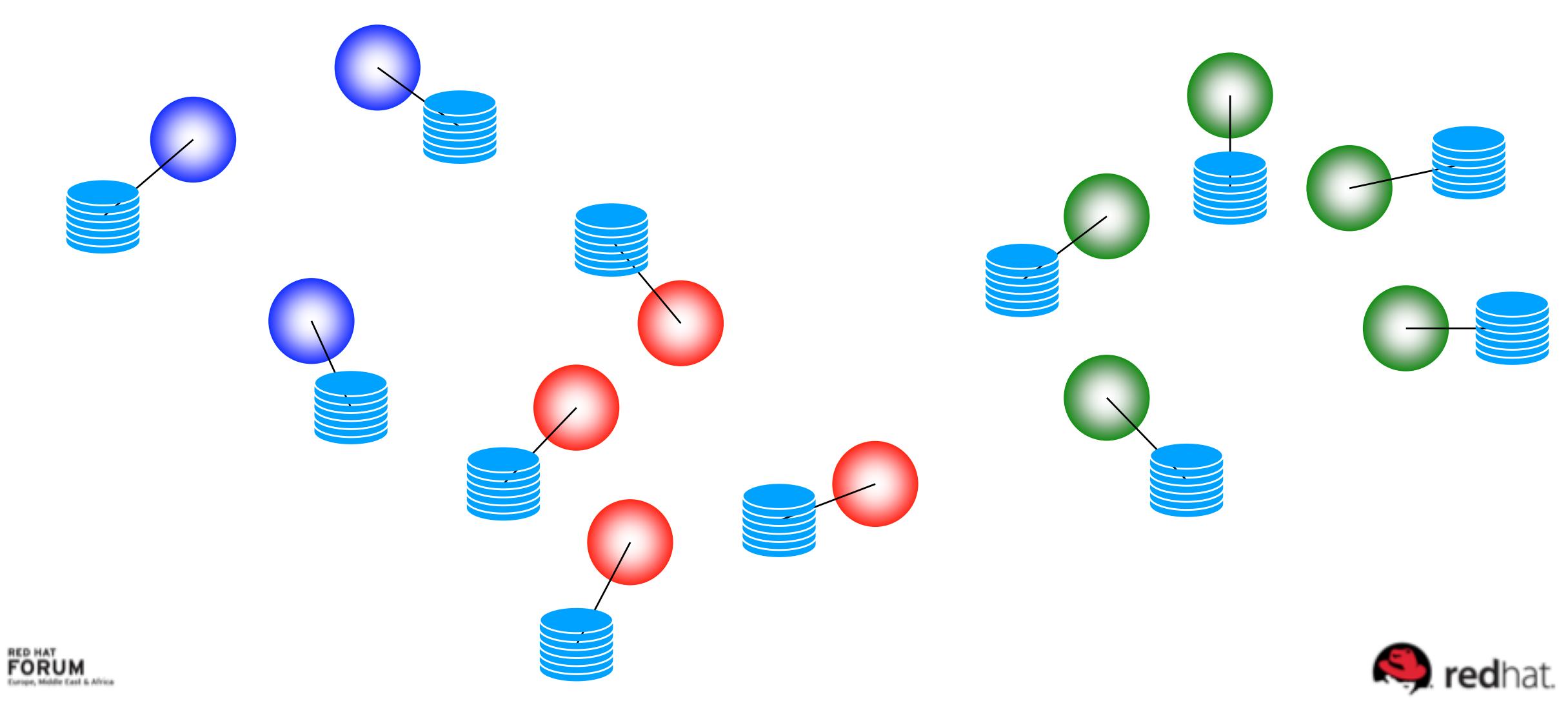




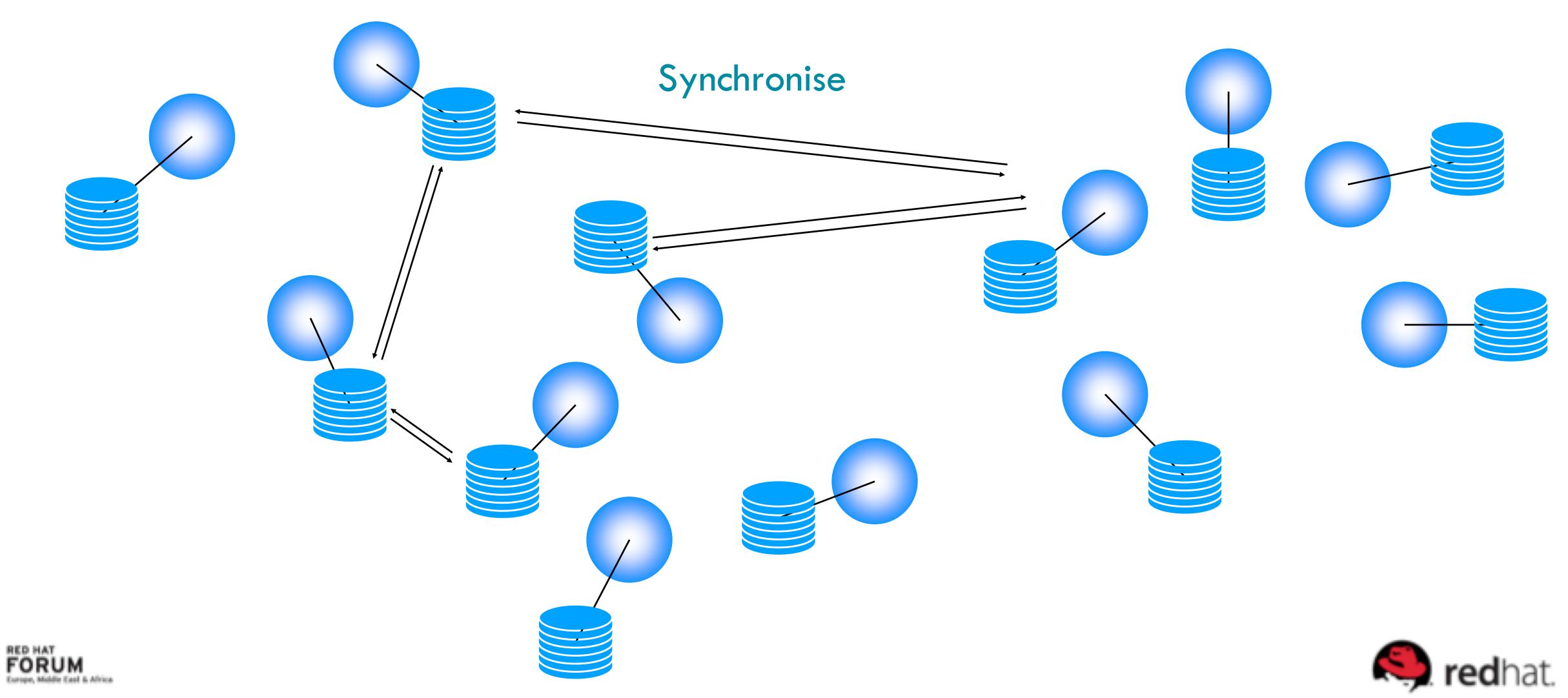


Data-centric

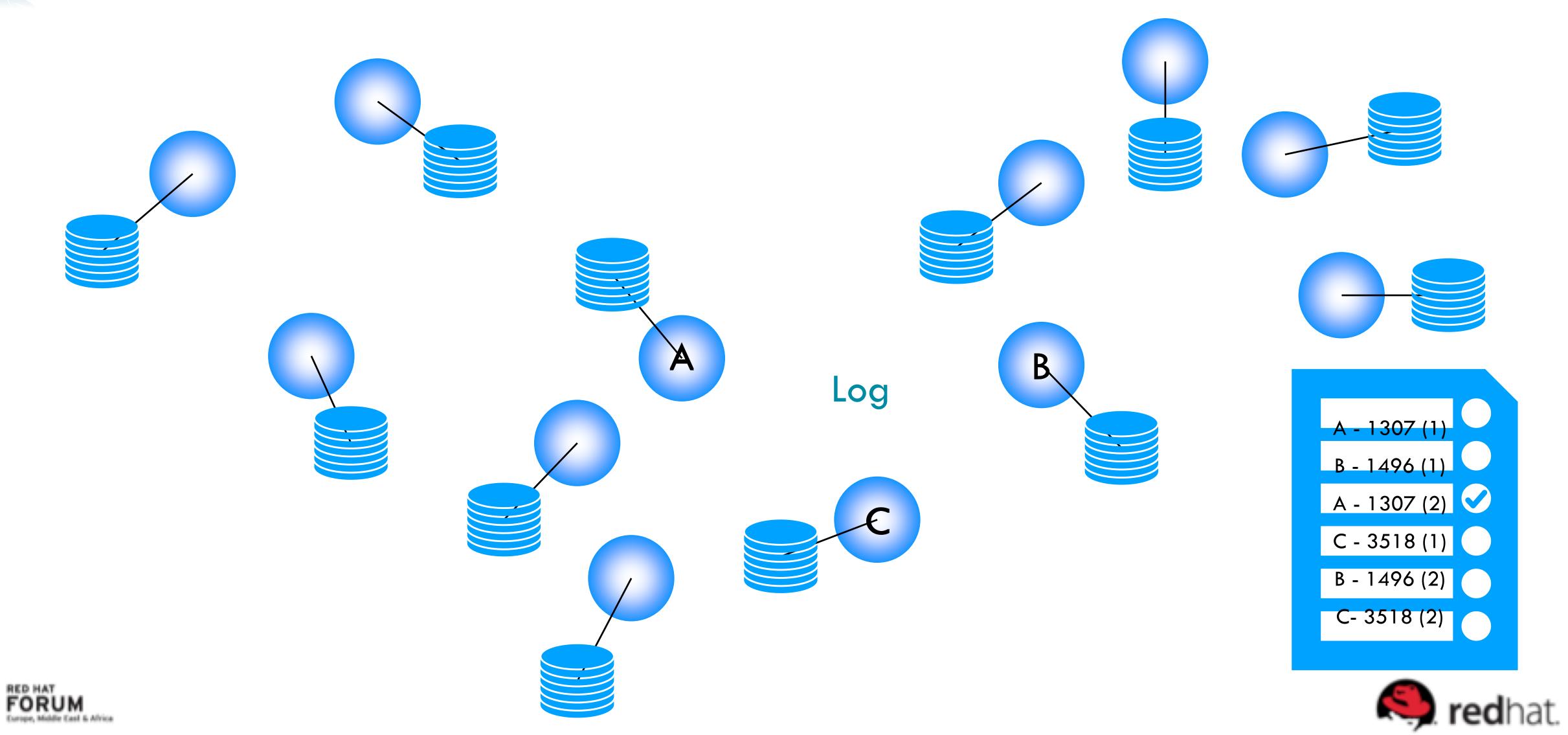




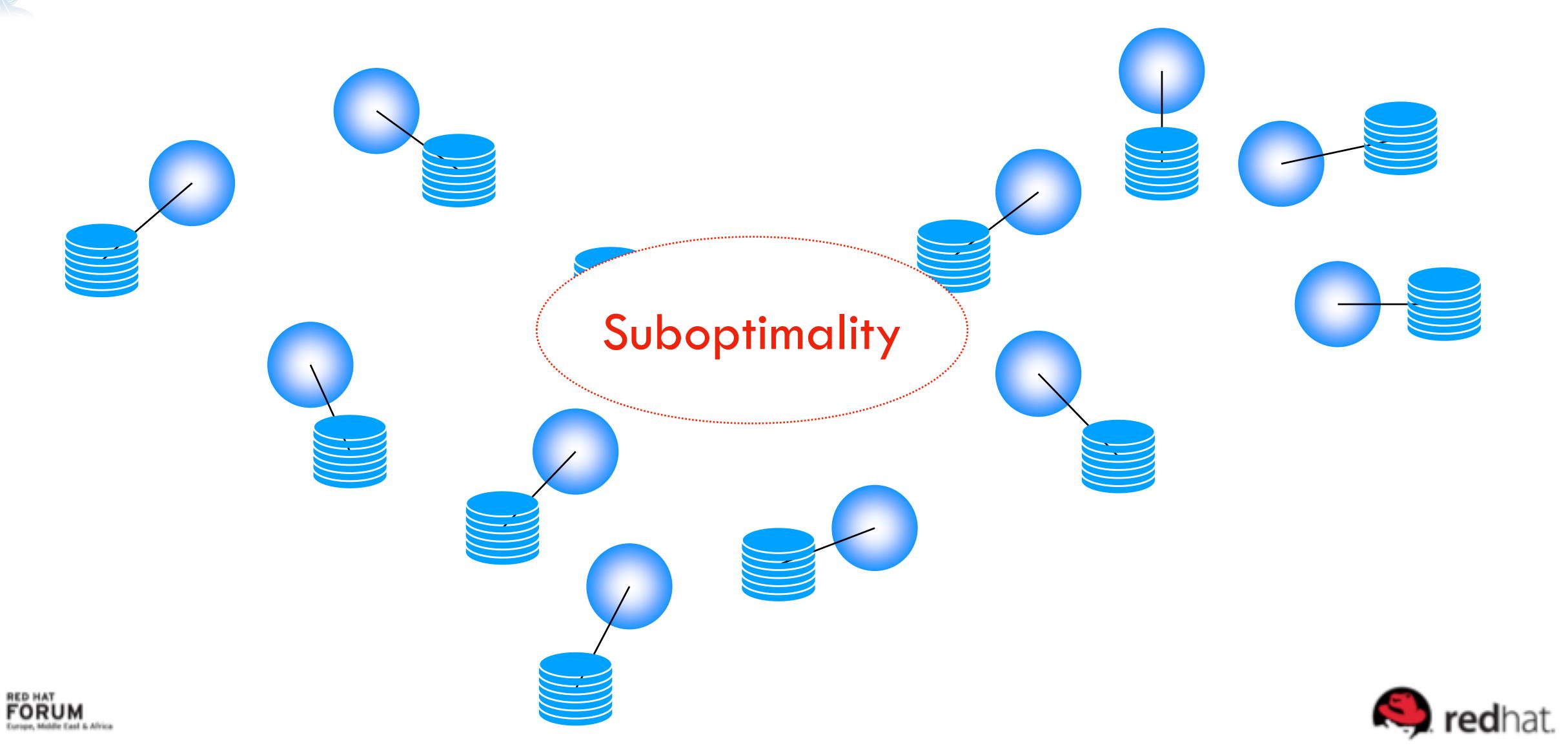












Quality

Rework

Suboptimality

Velocity

C121 R4 75 C122 C125 C135 C135 C141 C129 C135 C141 C135 C141 C129 C135 C141 C145 C14 C145 C1

C137

R178

0.0

a

RP72°

PH22

R P 6

3

R87

2 2222 2222

8.5

225

1111 4 2

RP34

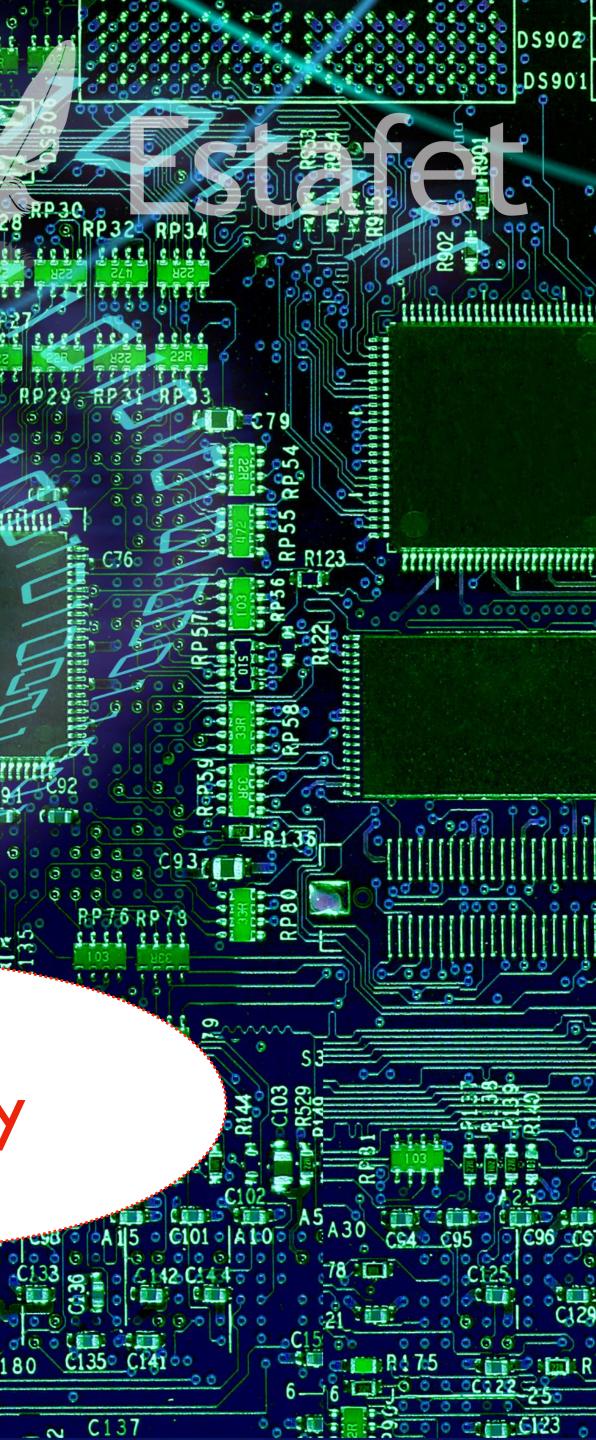
......

RP29

RP35

6PJ6 RP78

RP 2-6



€ R123

C15 00

Spotify[®]

Code first - ask questions later?

"Failures"



DIT M

NETELIX

What can we learn?

"Enough" up-front design

Design everything up-front

8

C121 : C123 C135 C141 C122 25 C123 C137 C123 C137

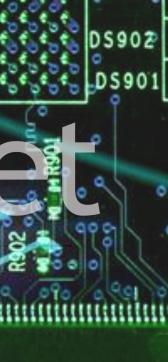
A30 CS4

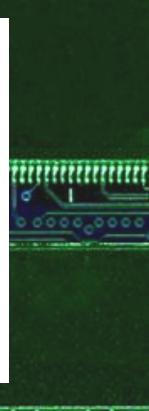
R178

5

C+48

RP50





A5 A30 CS4 C95 C96 CS 78 CS4 C95 C125

CI5 CI III: PA 75 C

C123 .

"Enough" up-front design

0123 4567 8901 2345



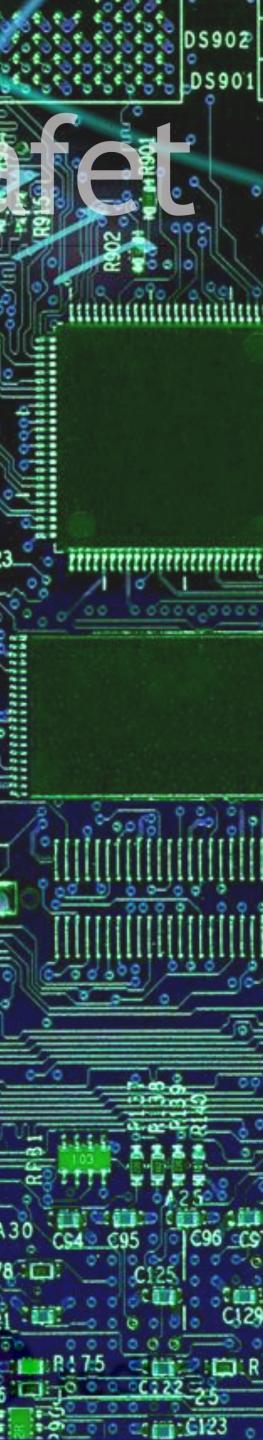


1. Write story cards to capture ...

2. Prioritise by ...

3. Plan release schedule

4. Scribble ...



(1) A5,

I.R180

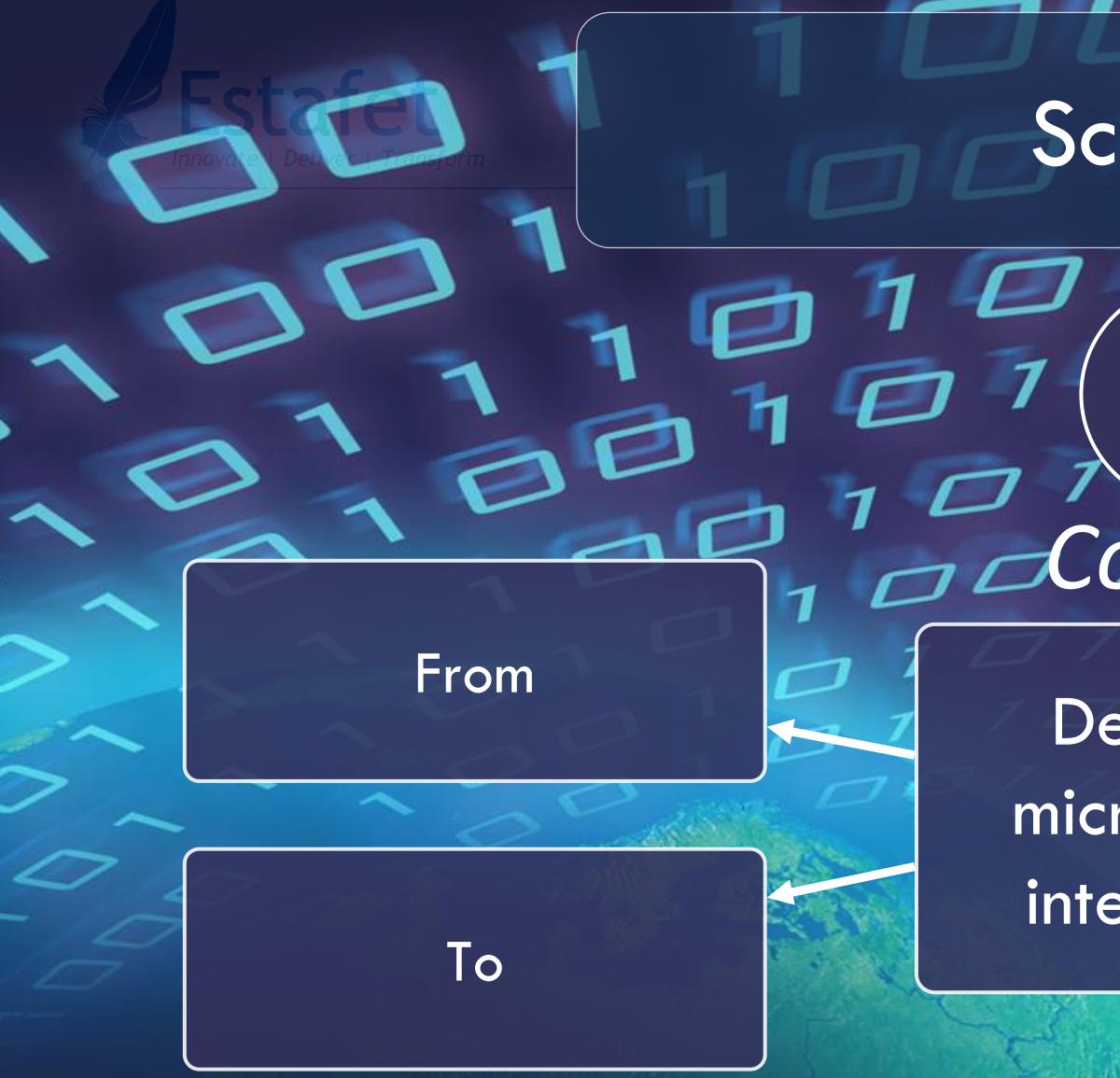
CG123 0

1 76 1 70 Cizz 25

C135° C141°

C137

- 2





Scribble

RP72

Control

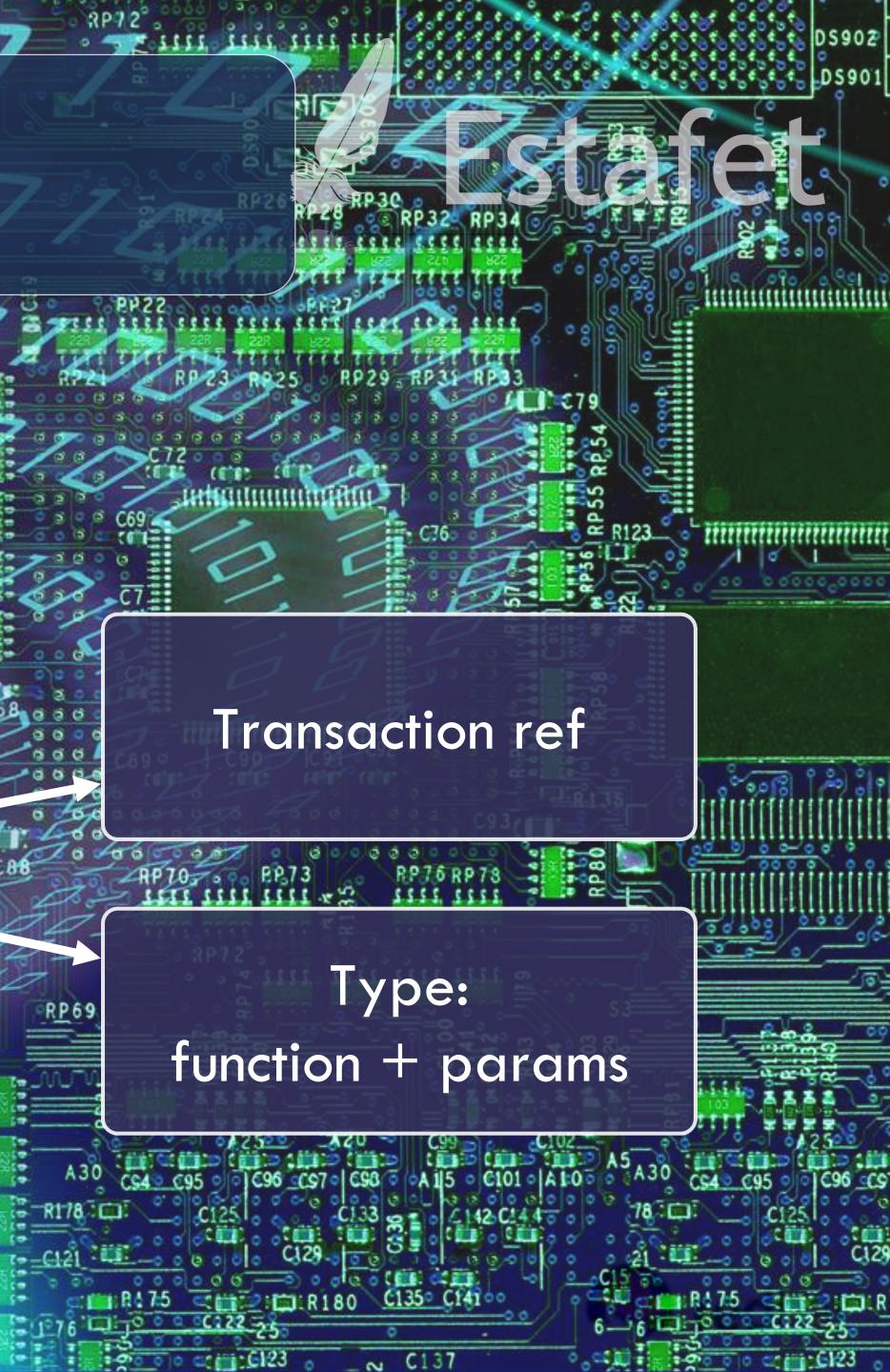
Describes microservice interactions

RP50

Transaction ref

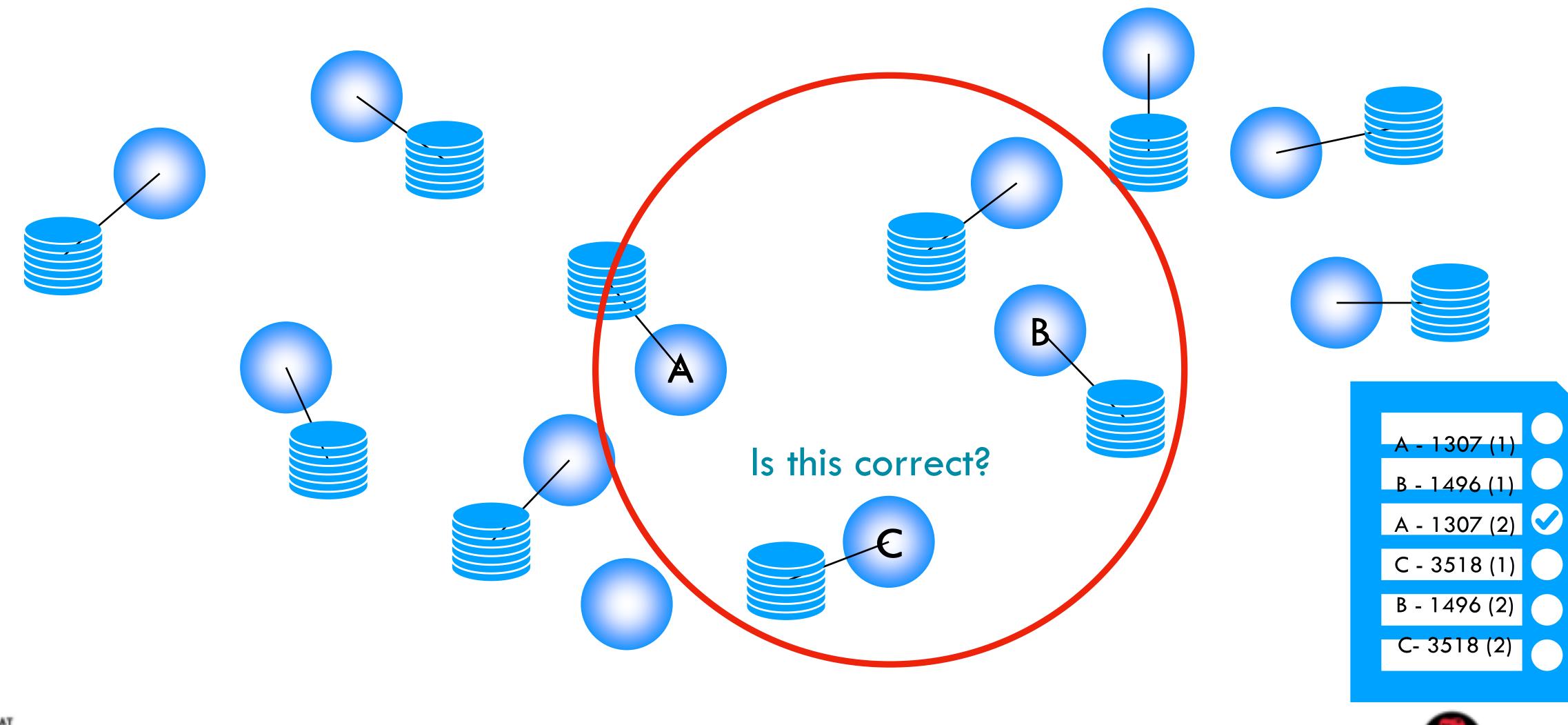
PP-73

Type: function + params

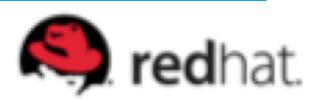




You cannot control what you do not understand





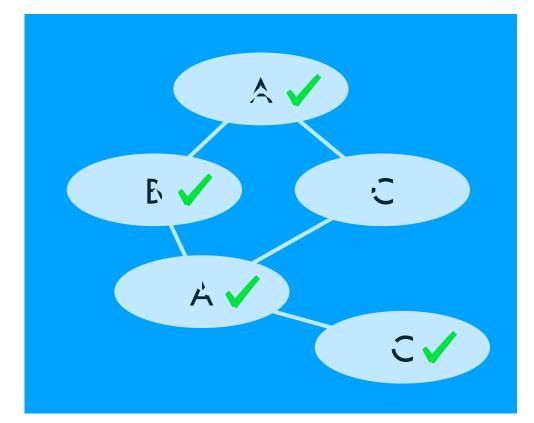








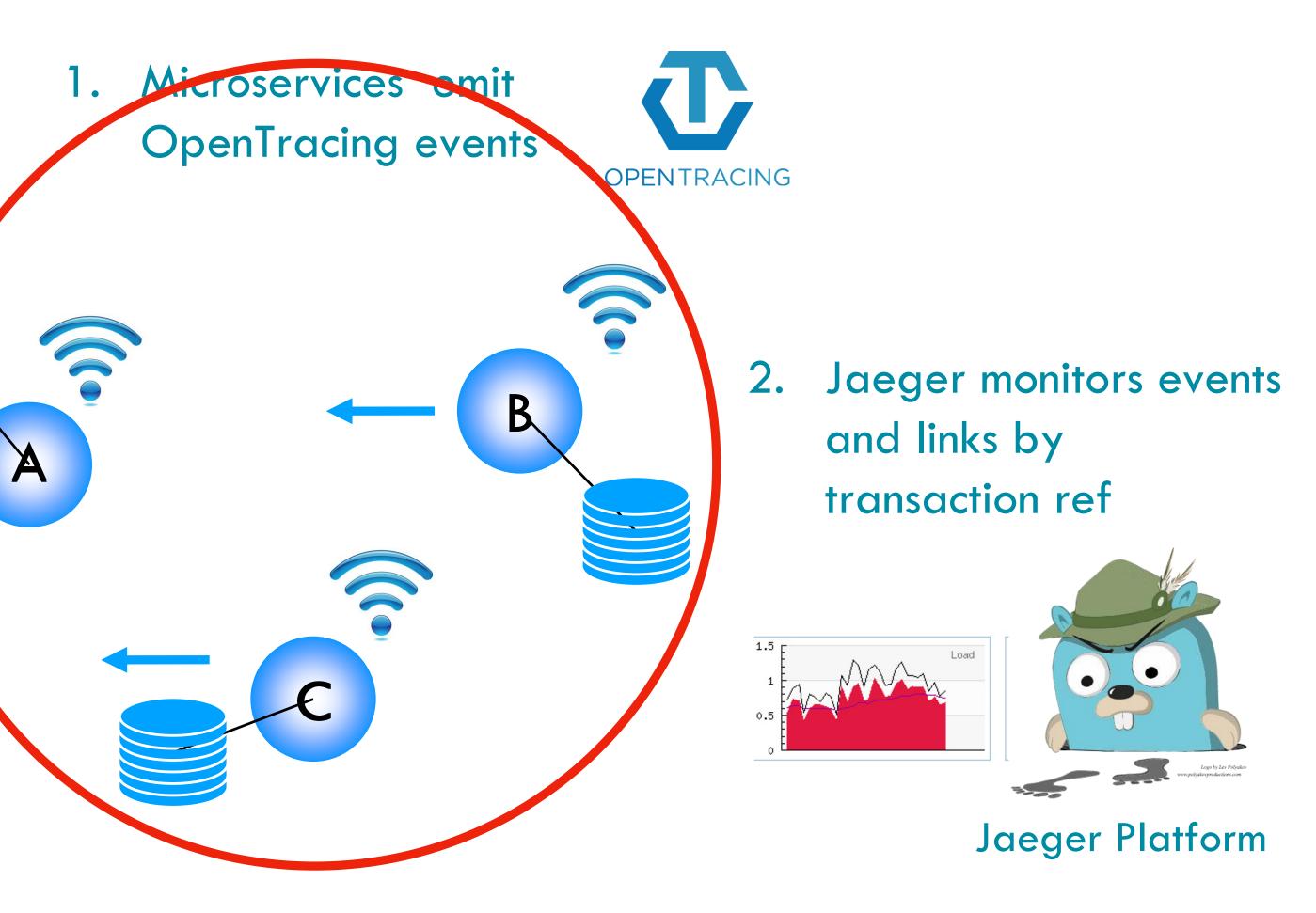
Scribble Model

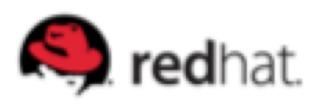


Complete 3. Check choreography is correct

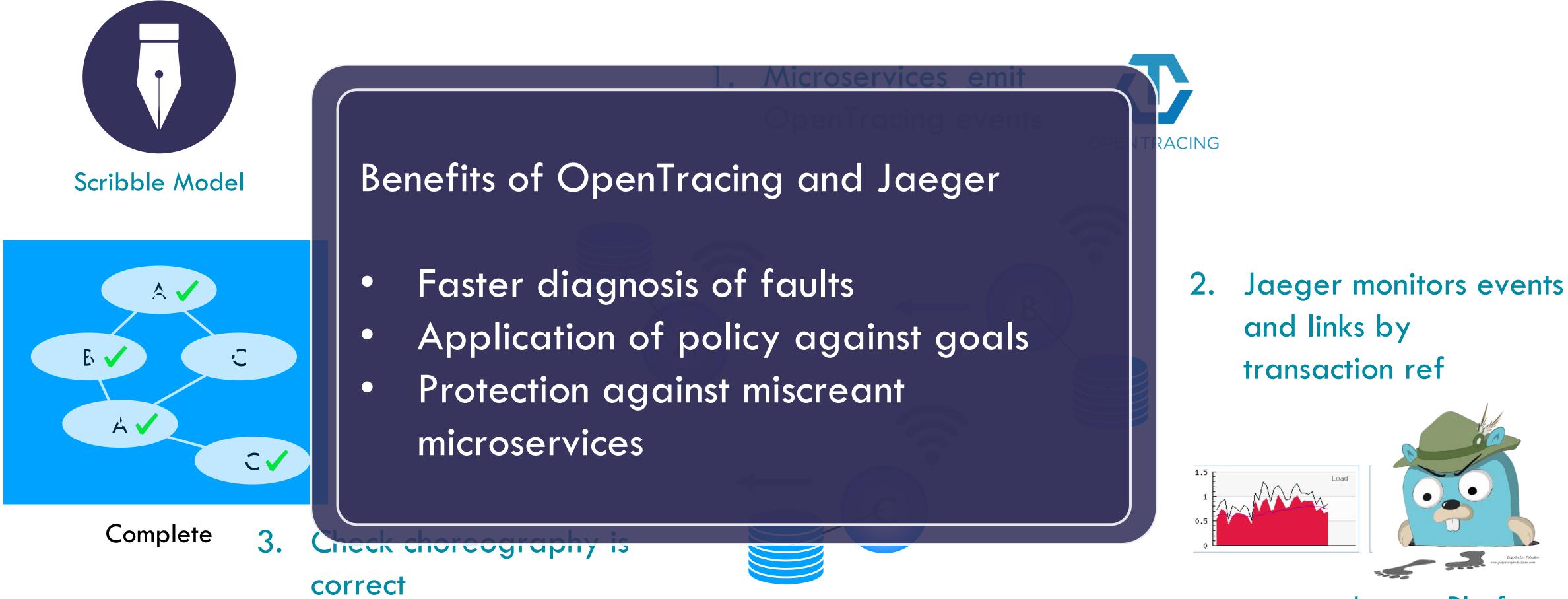


Monitor events to check behaviour











Monitor events to check behaviour

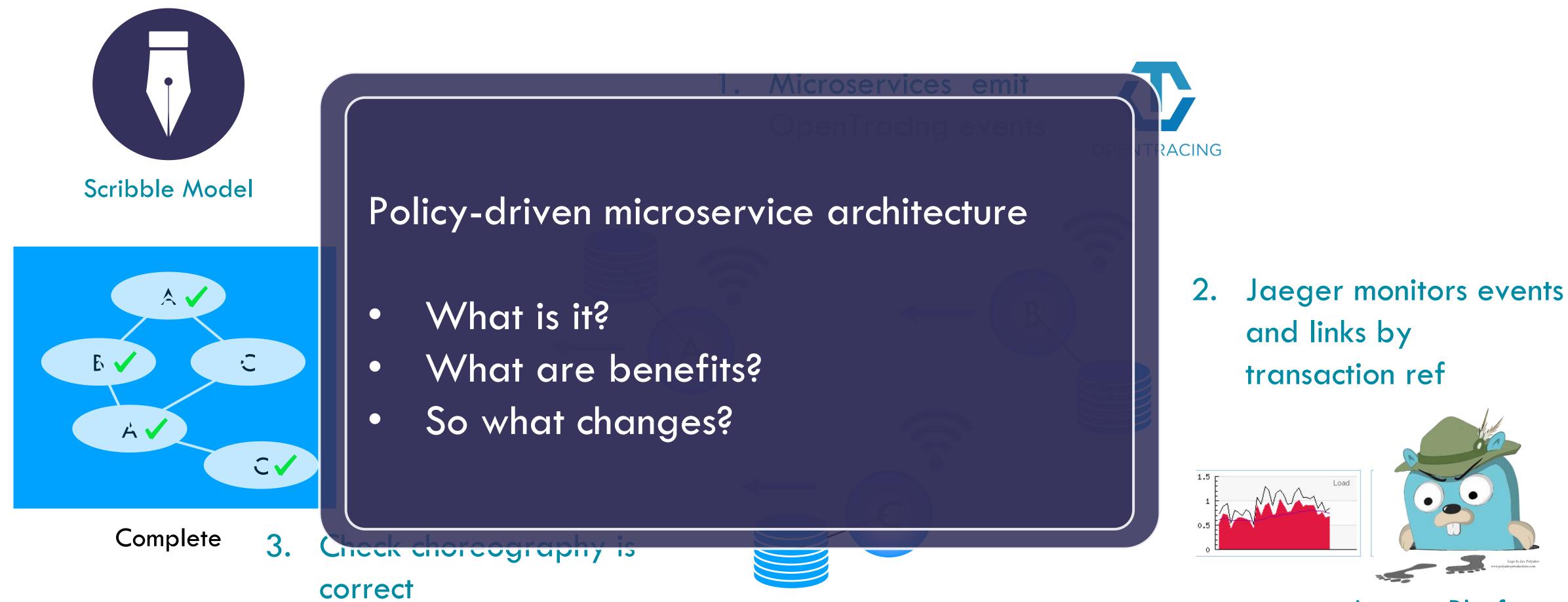
Jaeger Platform













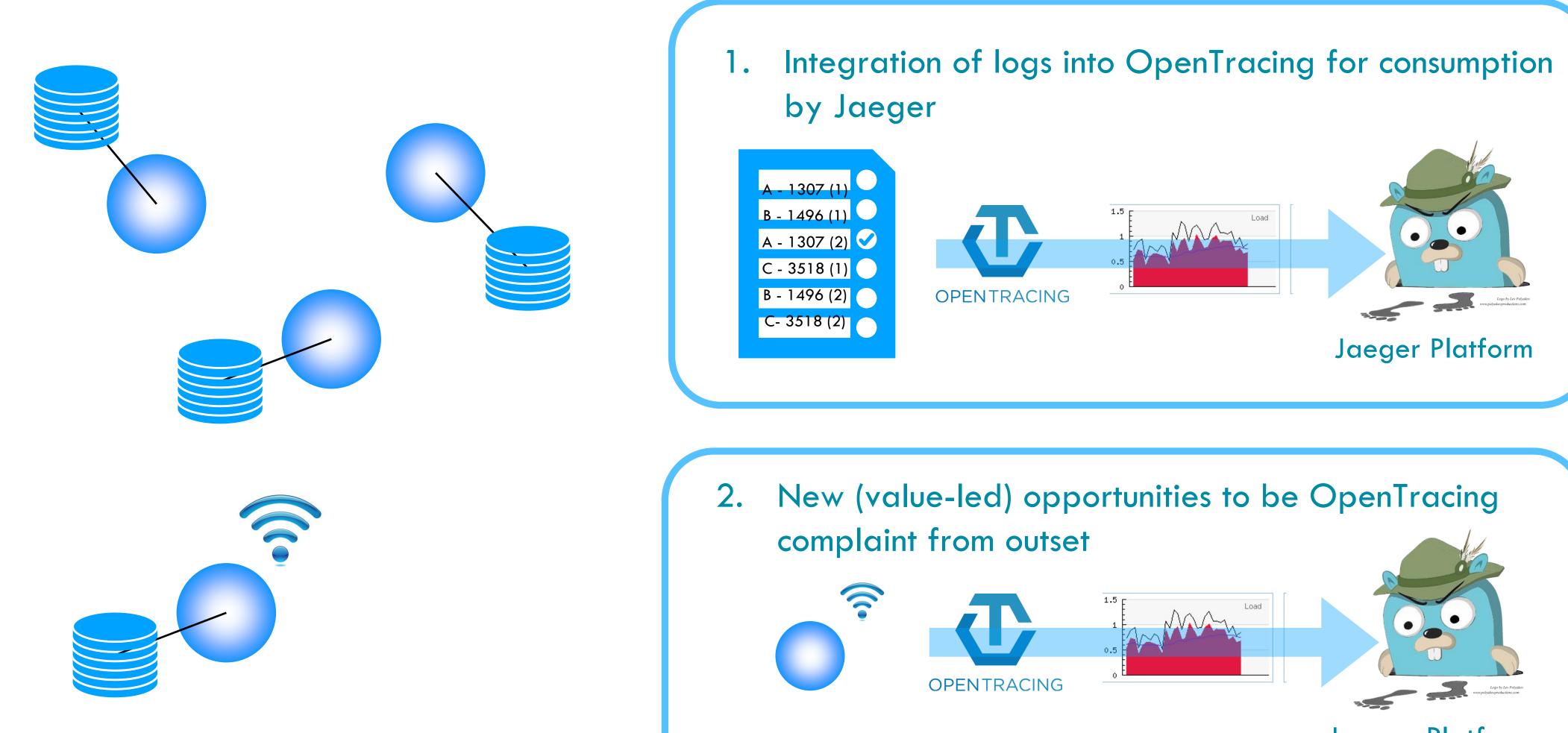
Monitor events to check behaviour

Jaeger Platform











What can I do now?

Jaeger Platform





redhat.



First Steps

- Two-week discovery to understand business flow and compare it with live logs;
- Share initial findings and then cost work required to create automated monitoring with SLA breach alerts;
- Build fast "root cause analysis" capability, leading to higher quality for lower cost.



What can I do now?

(COO)

Contacts

- <u>adrian.wright@estafet.com</u> (CEO)
- steve.ross-talbot@estafet.com (CTO)
- alistair.park@estafet.com

See our videos on

"<u>Why microservices fail</u>"

- goo.gl/GXgkkB
- "Managing distributed systems with Scribble"
- Estafet YouTube channel

<u>goo.gl/pzNeyG</u> <u>goo.gl/Vuiaxf</u>



