



Platforms with Product Thinking

Unlocking the power of platforms with a product mindset

Ed Seymour

EMEA EcoTech Sales SSA, Manager

Abstract

Faster to value and better customer experience with Platform as a Product delivery

Modern software development platforms are continually changing; Kubernetes has 3 releases a year that introduces new features and capabilities that developers are keen to start using. In addition, organisations are modernising their software engineering practice and delivery, but these transitions take years. Platforms need to accommodate the incremental shift from legacy software delivery to continuously improving cloud native apps, and in doing so allow for different teams going through this transformation at different rates.

Platform as a Product is platform design, build and run approach that treats the services as a continuously improving service, designed to meet the current needs of its users, whilst allowing for frequent releases and the introduction of new features. Whilst the approach requires a more sophisticated delivery than “run and maintain” approaches, it is more adaptive to customer needs and more efficient in meeting the need for change. Its responsive approach promotes improved customer relationships that create a stronger bond and support contract renewal.



Ed Seymour

EcoSystem Tech Sales SSAs

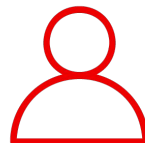


Application Platforms

A set of tools and services that developers use to build, deploy, and manage applications.

- ▶ Why invest in application platforms?
- ▶ The failure to keep pace with development
- ▶ Addressing the challenge
- ▶ Optimisation needs continuous improvement
- ▶ What we need to overcome to get there

It is a cross-cutting layer that ensures a consistent experience for acquiring and integrating typical capabilities and services for a broad set of applications and use cases.



[CNCF Platforms white paper.](#)

What the business wants

What is the implication of addressing the business goals



Speed & Efficiency

- Reduce overheads
- Increase productivity
- Faster to value

Security & Quality

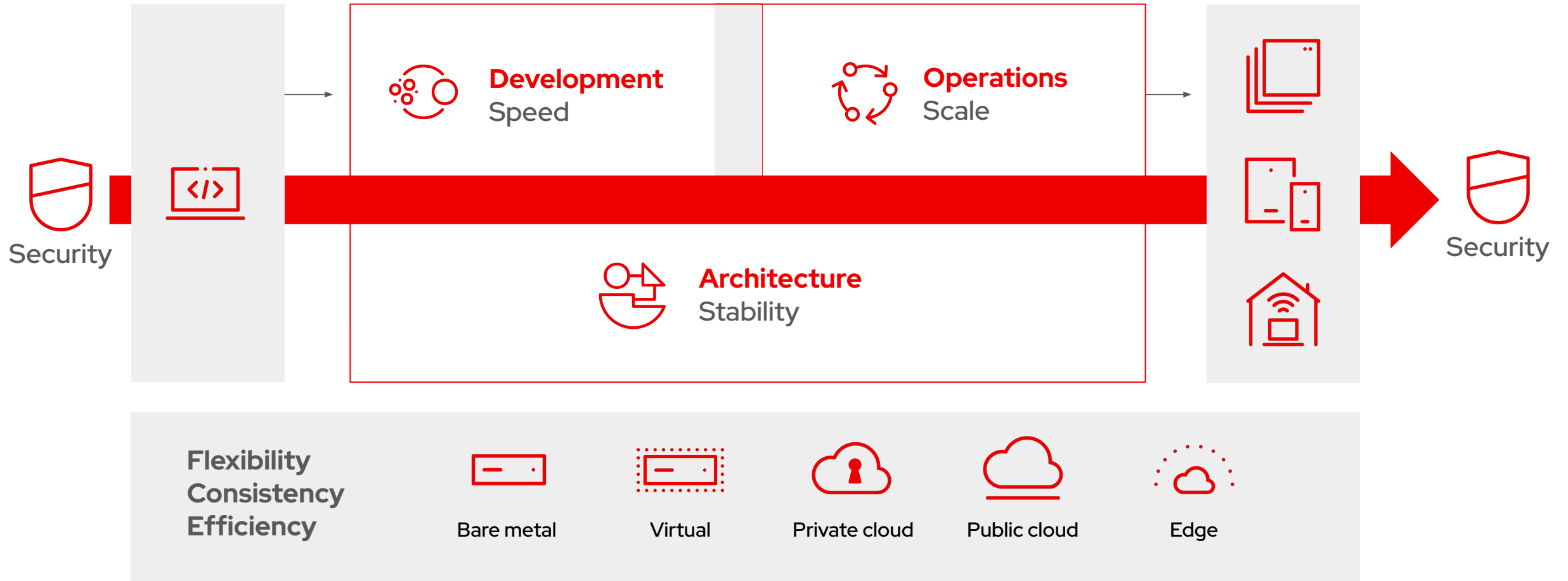
- Improve security
- Reduce errors in production
- Reduce time to resolution

Increased number of changes

**More changes
more often**

Higher frequency of change

What do Application Platforms need to do?



How they are being delivered today

Build up from infrastructure to simplify its complexity

Everything we think
we need for the next
3-5 years

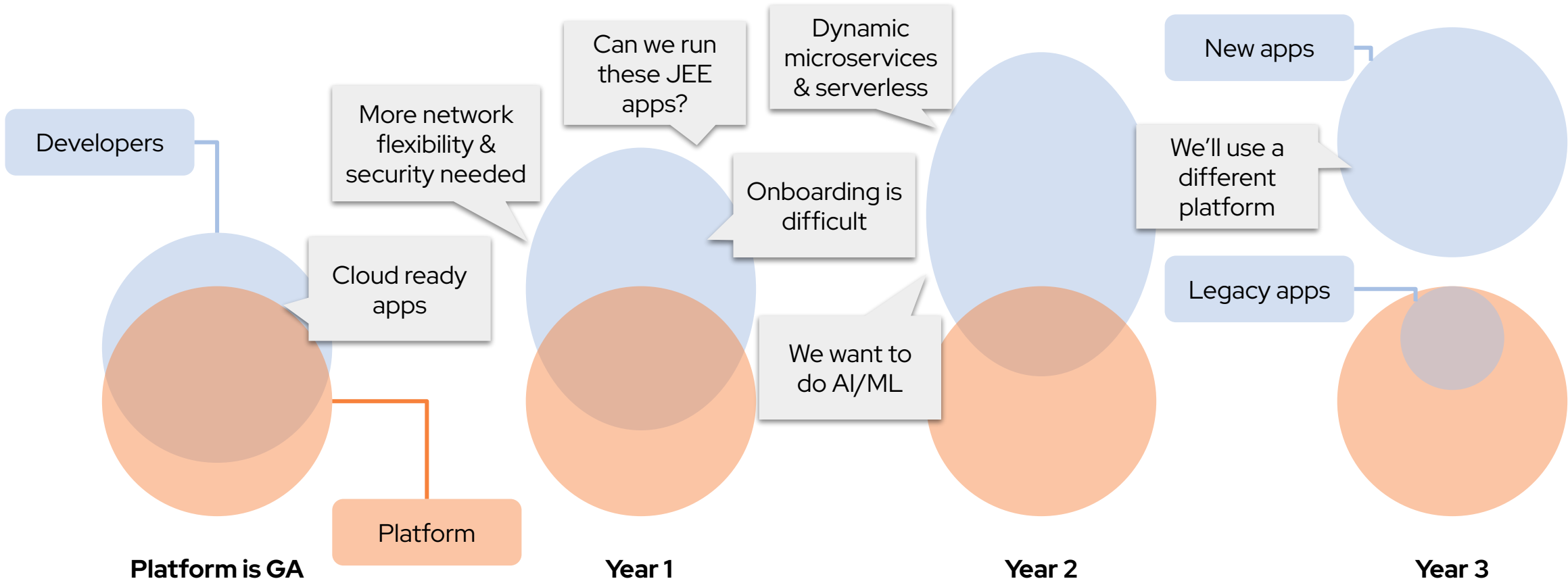


Upfront service build and long term maintenance & administration

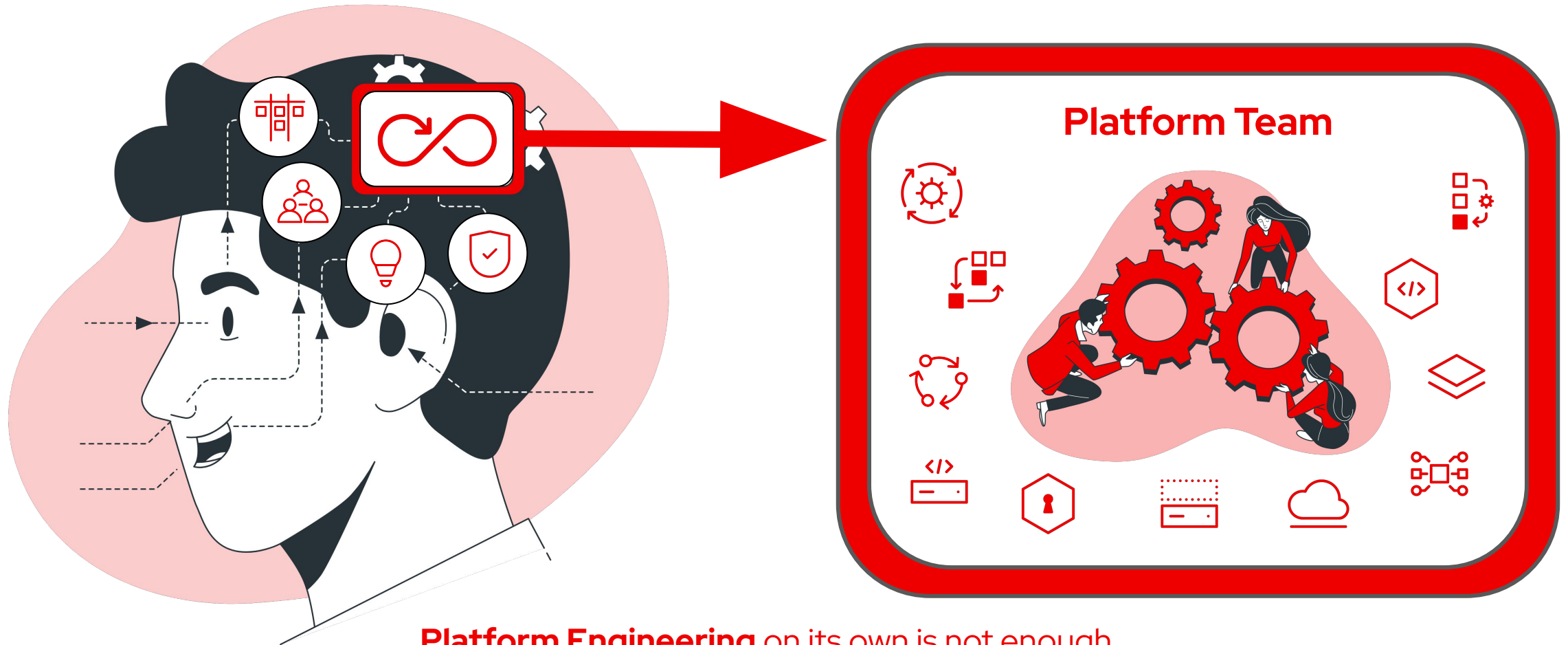
12-18 months

The trap of traditional platform delivery

An exercise in simplifying the underlying infrastructure



We need a developer centric platform delivery



Platform Engineering on its own is not enough

The platform team need to **respond to developer needs**

Evidence from Industry

Indicators from influential thought leadership organisations



"By 2026, 80% of software engineering organizations will establish platform teams ...

A dedicated **product team** creates and maintains the engineering platform."

- [Gartner](#)



"We keep getting good feedback from teams **applying product management** to internal platforms. ... We continue to see this technique as key to building internal platforms."

- [Thoughtworks](#)



"The paradigm shifts with Internal Developer Platforms. ... The platform team should view the internal developer **platform as a product** and look at developers as their customer."

- [Amazon](#)

Platform as a Product



Adopting a Platform as a Product approach

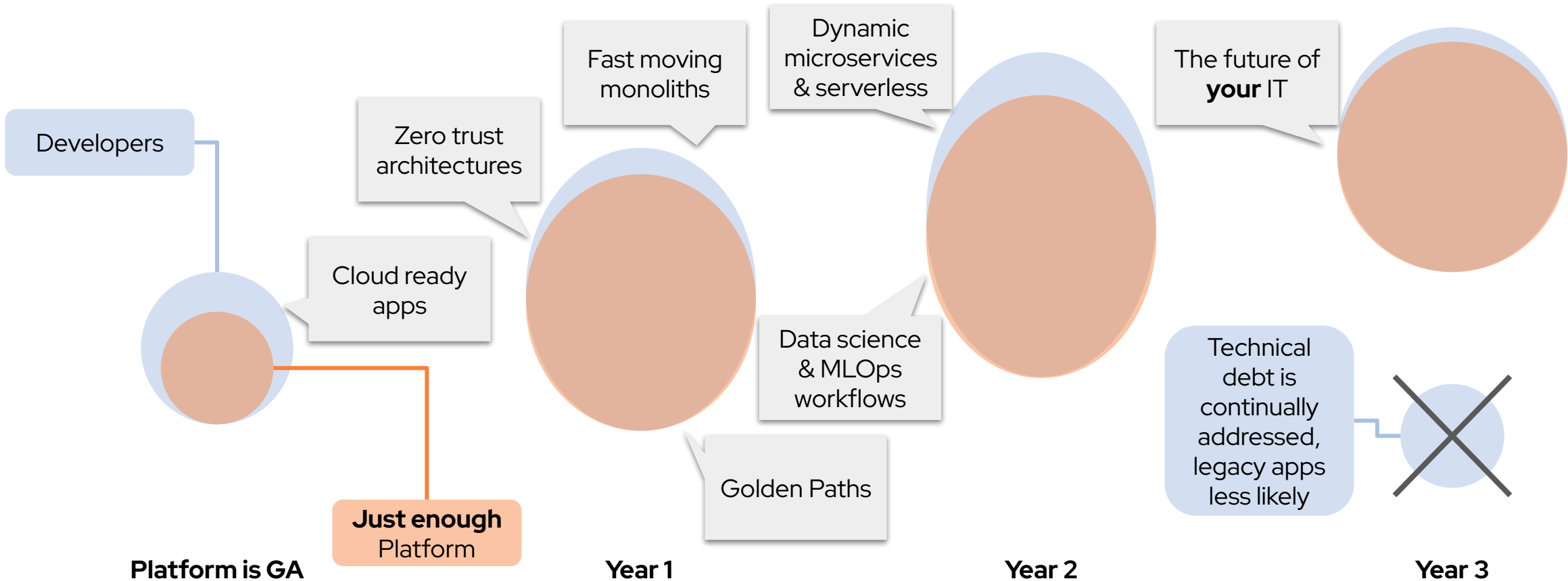
Create just enough platform to meet needs of the users

Continuous service improvement



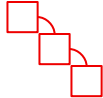









Continuously adapting platform

An exercise in addressing the specific needs of software development and delivery



Comparing with traditional approaches

Traditional delivery	Platform as a Product
 Maintaining a stable state	 Continuous improvement
 Requirements defined up front	 Developer centric & business needs driven
 Problems are remediated	 Problems are understood and engineered out
 Requests are handled via tickets	 Requests are managed automatically and on-demand
 Slow to adapt to changing needs	 Greater agility and incremental improvement aligned with user needs

Platform as a Product Benefits

Faster to value, reduced waste, lower risk, greater agility

What is stopping us?

The market expects design, build and run

Status quo This is how we've bought & built services in the past

Vicious circle How we go to market perpetuates the status quo

Scope creep Yes, but I need this in release 1

Happy space Engineers just be engineering

Engineering investment

Platform as a product mitigates risk and promotes more effective adoption

Platform is GA

Year 1

Year 2

Year 3

Year 4

Why should we care?

What is the value of adopting a Platform as a Product approach?

The business

Faster to value & continuously optimising service

The developers

Provides necessary services & doesn't get in the way

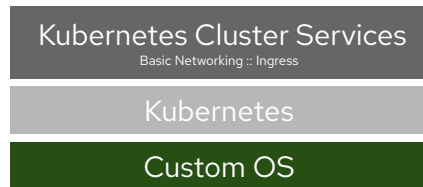
The platform teams

Avoid becoming a cost centre by continually adding value

Red Hat OpenShift accelerates the Platform Engineers



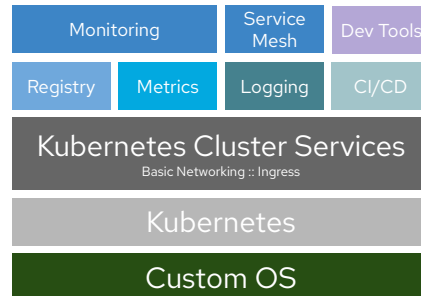
The Engine



xKS



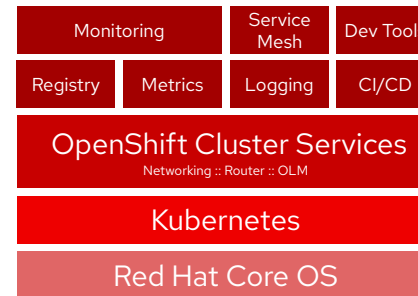
The Parts



xKS
plus 'native' services



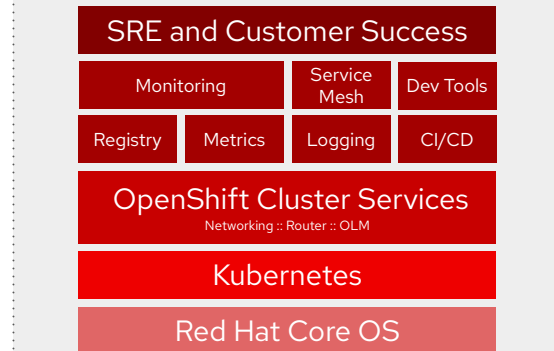
The Assembled Car



Self-managed
Red Hat OpenShift



The Full Service



Red Hat OpenShift Service on AWS
Azure Red Hat OpenShift

Platforms with Product Thinking



Business drivers inevitably lead to increased change



Platforms need to **optimise** with development

A static platform becomes increasingly less useful



Traditional platform delivery cannot adapt quickly enough



Improve with **developer centric** platform engineering



Accelerate the platform engineers with Red Hat

What's next?

1 Do you see value in this approach?

2 What's preventing you from doing this?

3 Join the Platform as a Product meet up!

Red Hat is hosting the [first event in London, Nov 23rd](#)

Thank you



Thank you



linkedin.com/company/red-hat



facebook.com/redhatinc



youtube.com/user/RedHatVideos



twitter.com/RedHat