



SUMMIT
BERLIN

Think you are ready for Microservices?

Markus Kaiser
Solutions Architect
AWS

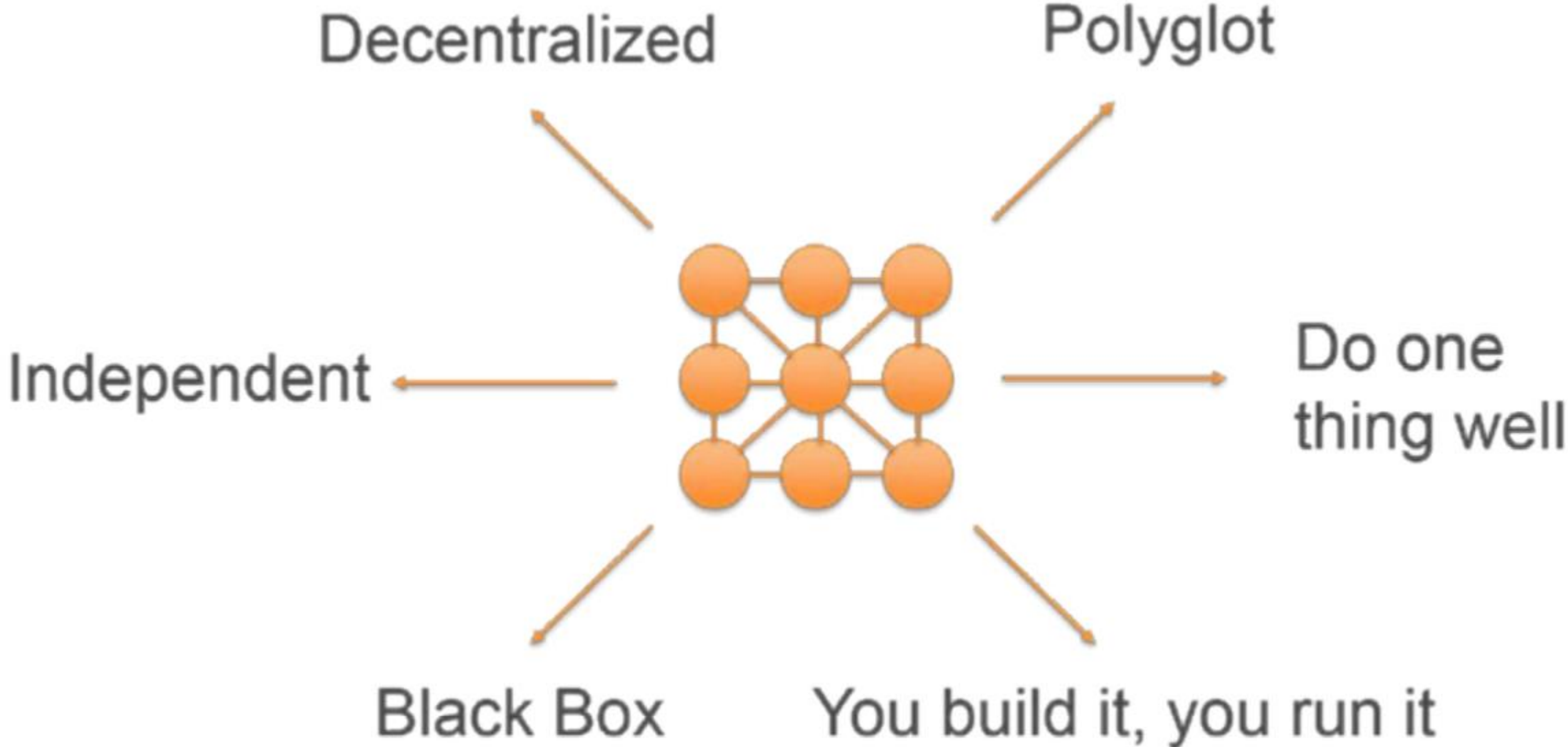
Kilian Ruess
Solutions Architect
AWS

What to expect from this session

- > Key Business Drivers for Microservices Adoption
- > Expectations & Challenges
- > How to start, today? Best-Practices

Microservices

Characteristics of Microservices

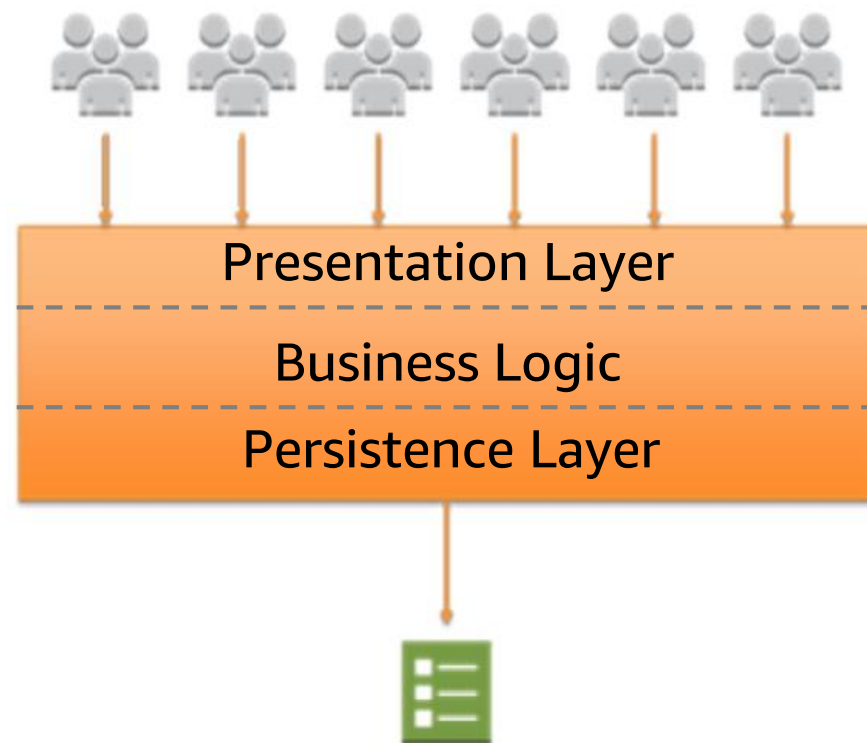


Agility

Ability to move quickly & change directions with ease

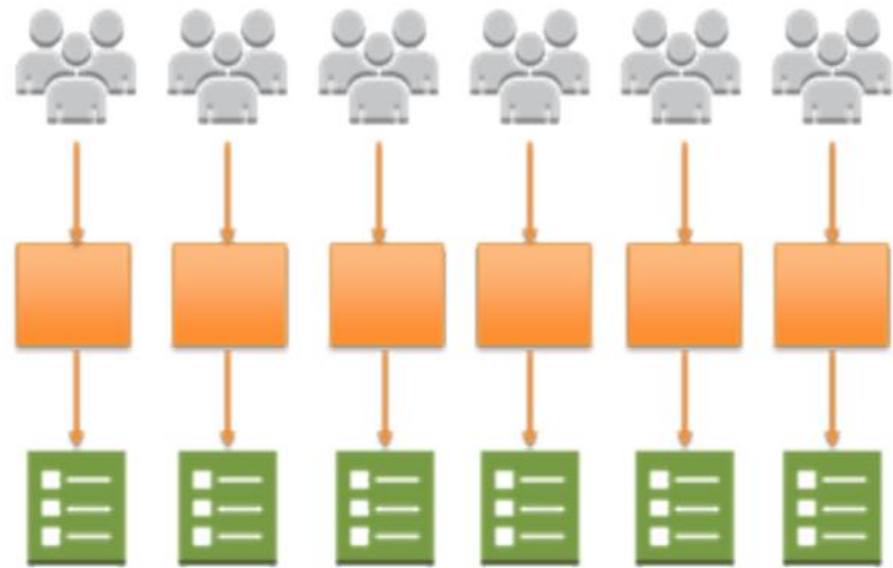
Monolithic Applications

Example: 3-tier-application



- Functional teams are responsible for different layers
- Business requirement involves multiple layers, i.e. multiple teams
- Common deployment
- Quarterly or annually release cycles

Microservices & Agility



- **Do one thing well** – Requires knowledge of a small, well-understood domain.
- **Independent** – Empowers small, independent teams to move at their own pace → shortening the cycle times
- **You build it; you run it** – Fully-responsible incl. QA, deployment, and production

Challenges

Do one thing well – Requires knowledge of a small, well-understood domain.

Break-down the monolith along meaningful domain boundaries

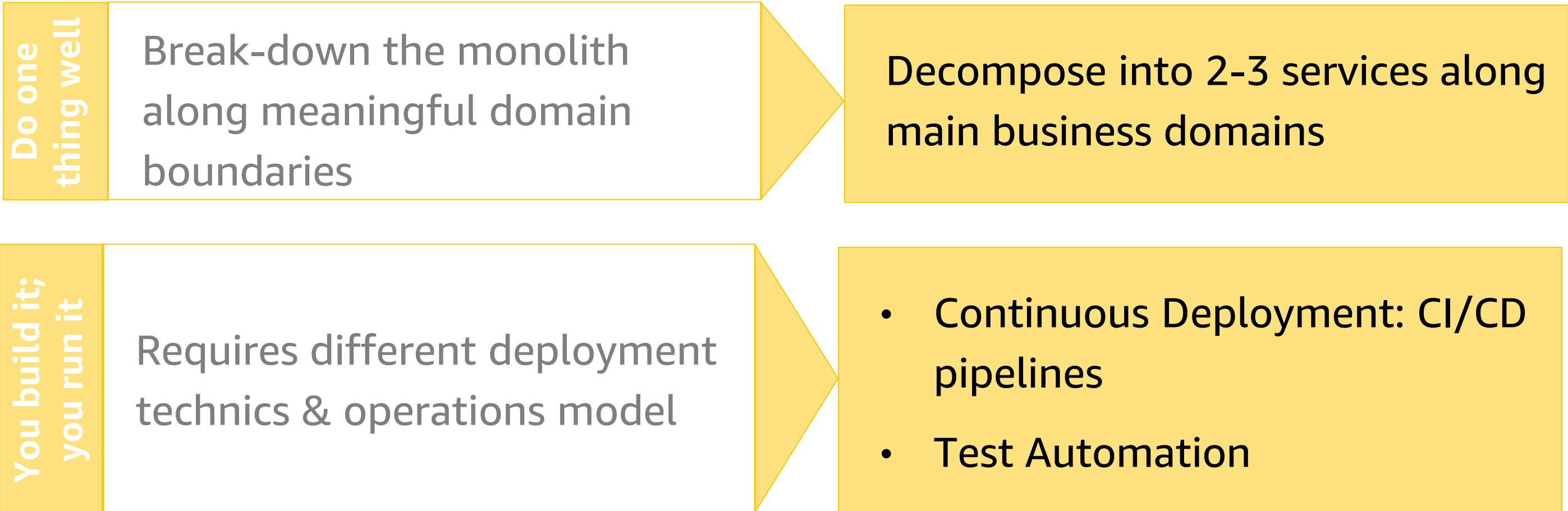
Independent – Empowers small, independent teams to move at their own pace → shortening the cycle times

- Requires Cross-Functional Team with all necessary skills
- Increased Architectural and Operational Complexity

You build it; you run it – Fully-responsible incl. QA, deployment, and production

Requires different deployment technics & operations model

Where to start?



How AWS can help?

Separation of Concerns

- AWS Features, e.g. Multi-Account Strategy

Continuous Deployment

- AWS Services for CI/CD
- On-Demand Resources & Infrastructure Automation

Architectural & Operational Complexity

- Managed Services, Serverless?
- Architecture Review by Solutions Architect

Innovation

Microservices & Innovation

Do one thing well – Vertical alignment with the business owners

Independent – Fast-feedback loops & low cost of failure

Decentral & Polyglot – freedom to choose and replace the technology stack individually (per domain)



Challenges

Do one thing well – Vertical alignment with the business owners

Cultural transformation, from hierarchical decisions to empowered teams

Independent – Fast-feedback loops & low cost of failure

Cultural of experimenting & failure tolerance needed

Decentral & Polyglot – freedom to choose and replace the technology stack individually

Synergy effects such as sharing libs and people get lost

Where to start?

Do one
thing well

Cultural transformation, from hierarchical decisions to empowered teams

- Retreat governance structures
- Shift business model to product strategy

Independent

Cultural of experimenting & failure tolerance needed

- Identify business KPIs to measure success / failure
- Embrace change & experiments, evaluate the risks

How AWS can help?

Culture of Innovation - Learn from Amazon's experience and culture: Working backwards, two-way doors, etc.

Technological Innovation - Benefit from the breadth of AWS services and AWS pace of innovation

Experiment with low cost and risk – On-demand, pay-per-use reduces the cost of experimenting with new ideas.

Scalability

Microservices & Scalability

- **Independent** – Properly decoupled services can be scaled horizontally and independently from each other
- **Decentral** – leverage the optimal data persistence solution per service
- **Polyglot** – appropriate and optimal technologies for a specific service

Challenges

Independent – Properly decoupled services can be scaled horizontally and independently from each other

- Put individual scalability measures in place
- Identify scalability bottlenecks in overall system

Decentral & Polyglot – leverage the optimal technologies for a specific service

Overall system scalability difficult to understand

Where to start?

Classic

Follow classical approach for distributed systems

- If needed break monolith along different scalability requirements
- Employ classical options for horizontal scaling

Scalability Drivers

Make scalability tangible to enable focused architecture decisions

- Identify & measure scalability drivers
- Know your limits, perform scalability tests

How AWS can help?

AWS Services

- Auto-Scaling Features
- Monitoring & Automatic Response
- Serverless?

Quality & Availability

Microservices & Quality

- **Do one thing well** – Reduced complexity
- **Independent** – Leads to *smaller deployments & failure isolation*, and allows *graceful failure handling*
- **You build it; you run it** – Fully responsible teams

Challenges

Do one thing well – Reduced Complexity

Increased infrastructure complexity introduces new sources of failure (Networking, etc.)

Independent – Graceful failure handling

Graceful failure handling is not for free

Monolithic vs Microservices



Monolithic



Microservices

“If you can’t build a well-structured monolith, what makes you think microservices is the answer?”

Simon Brown

Where to start?

Do one
thing well

Increased infrastructure complexity → introduces new sources of failure

Continuous Deployment

Independent

Graceful failure handling is not for free

- Architect for Failure
- Build a Learning Organization:
e.g. Root Cause Analysis & Lesson Learned process

How AWS can help?

Continuous Deployment

- AWS Services for CI/CD
- On-Demand Resources & Infrastructure Automation
- Features for Deployment Strategies such as blue/green, canary, etc.

Wrap-Up

Summary

Microservices

- are technically demanding
- have impact on organization & people
- emerge / benefit from a certain company cultural

Think you are ready for Microservices?

Thank you!

Markus Kaiser
@markuskaiser
kamarkus@amazon.de

Kilian Ruess
@kilianruess
kruess@amazon.de



Please complete the
session survey.