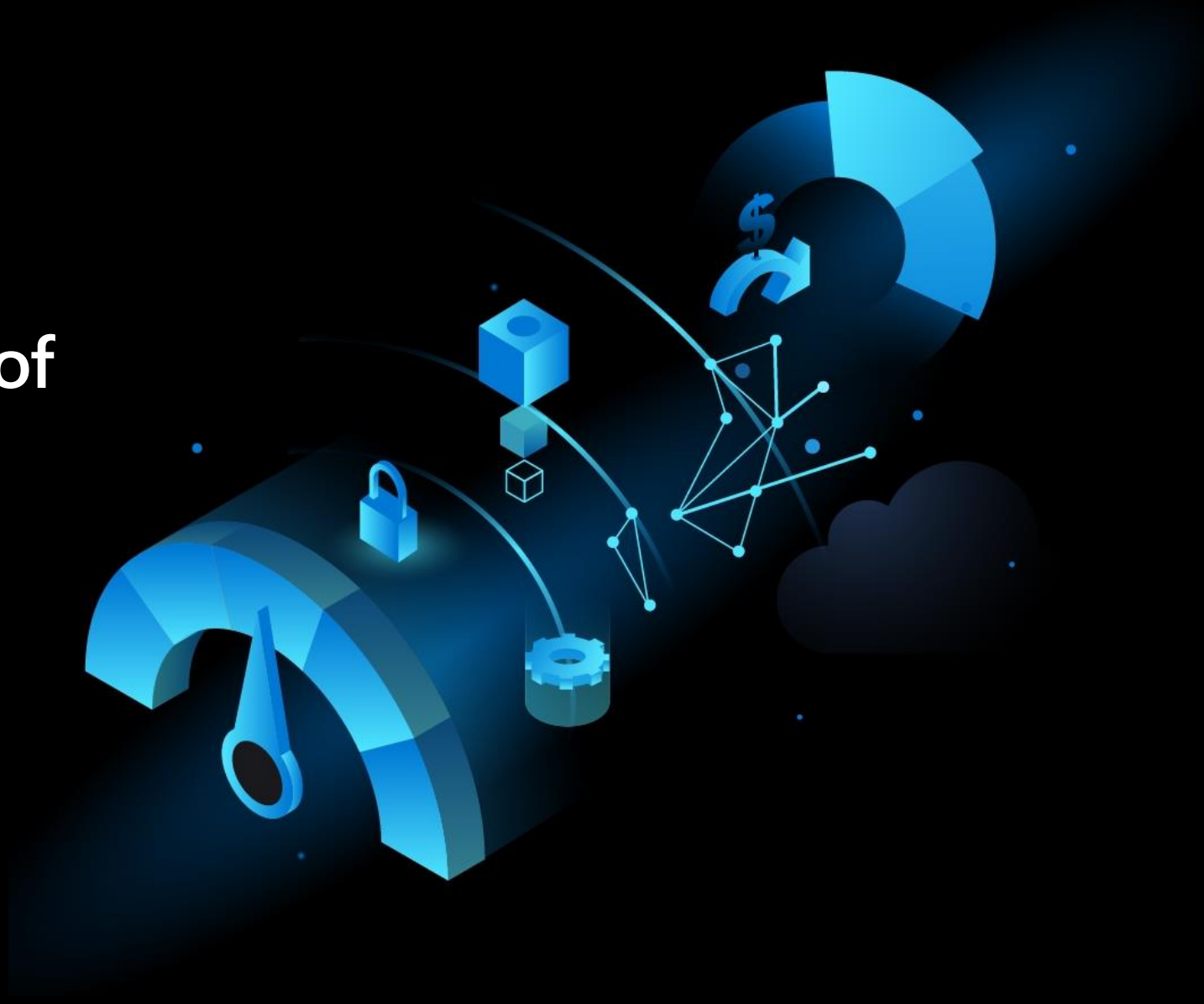
Microsoft

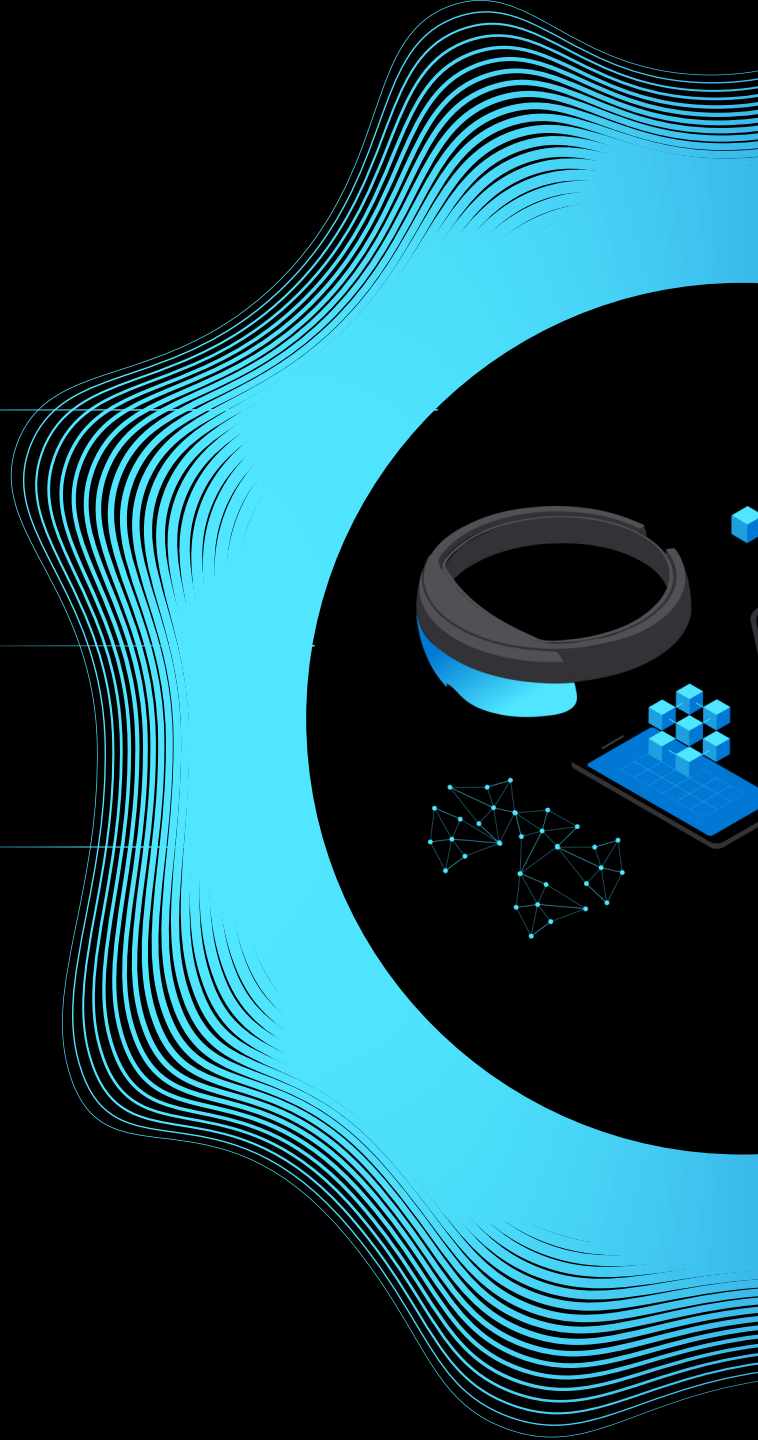
Test your changes as part of your pipeline



Gerard van de Pol & Remco Eissing
Sr. Specialist & Cloud Solution Architect

Agenda

- Continuous testing
- Load Testing
- Chaos Engineering

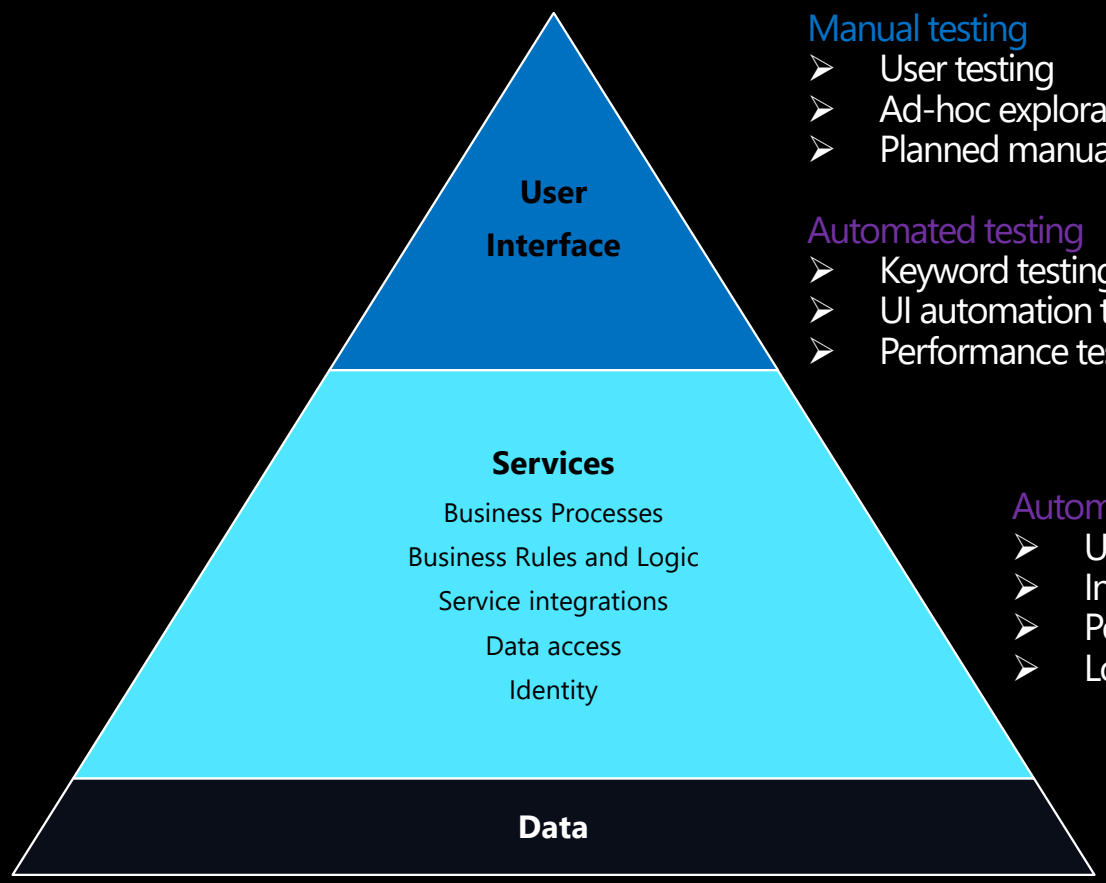


Why Quality Assurance matters?

Quality Assurance is traditionally done at the advanced development stage

DevOps Shift-left mentality

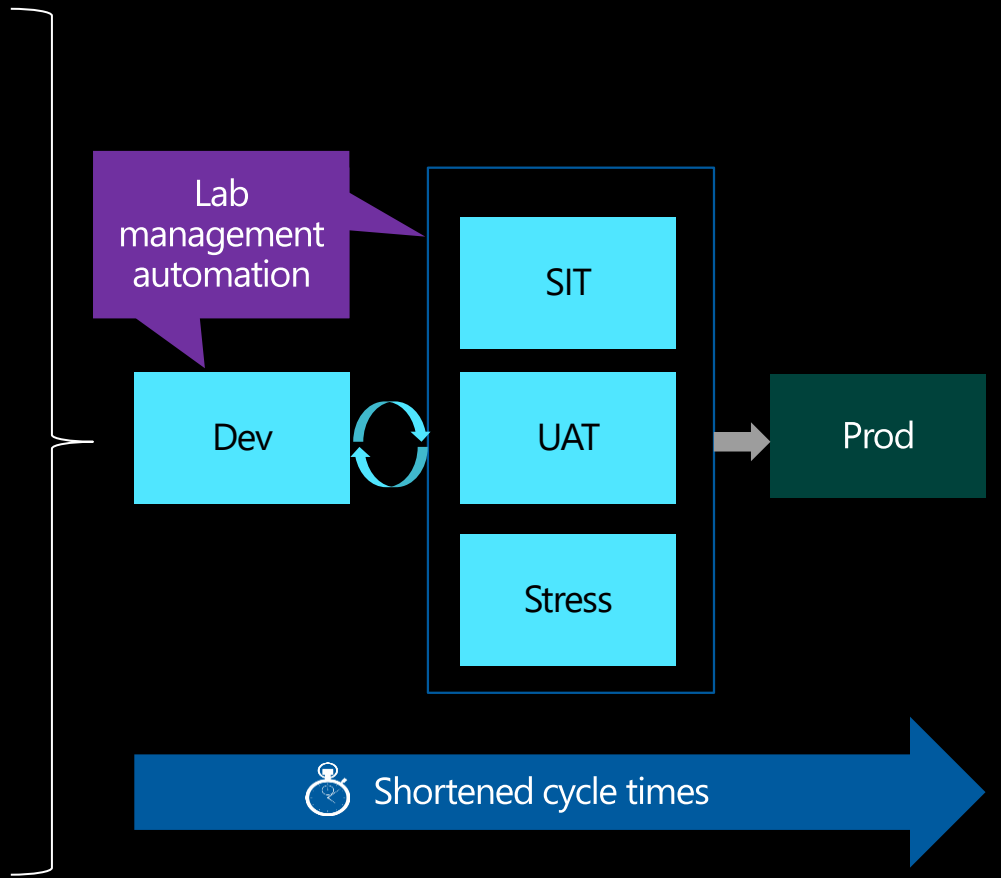
Continuous acceptance testing



Manual testing
- User testing
- Ad-hoc exploratory testing
- Planned manual testing

Automated testing
- Keyword testing
- UI automation testing
- Performance testing

Automated testing
- Unit testing
- Integration testing
- Performance testing
- Load testing



Can my code handle the load?

Validate with load test and adjust

Late and breaking architecture and design changes are costly

Early testing, adjusting based on data

Continuous adjustments to design and architecture to handle the scale

Performance testing definitions

Load testing – simulated load, measuring overall performance

Performance testing – measure performance regardless of load

Stress testing – simulate load to the system until it fails.

Traditional Load Testing Challenges

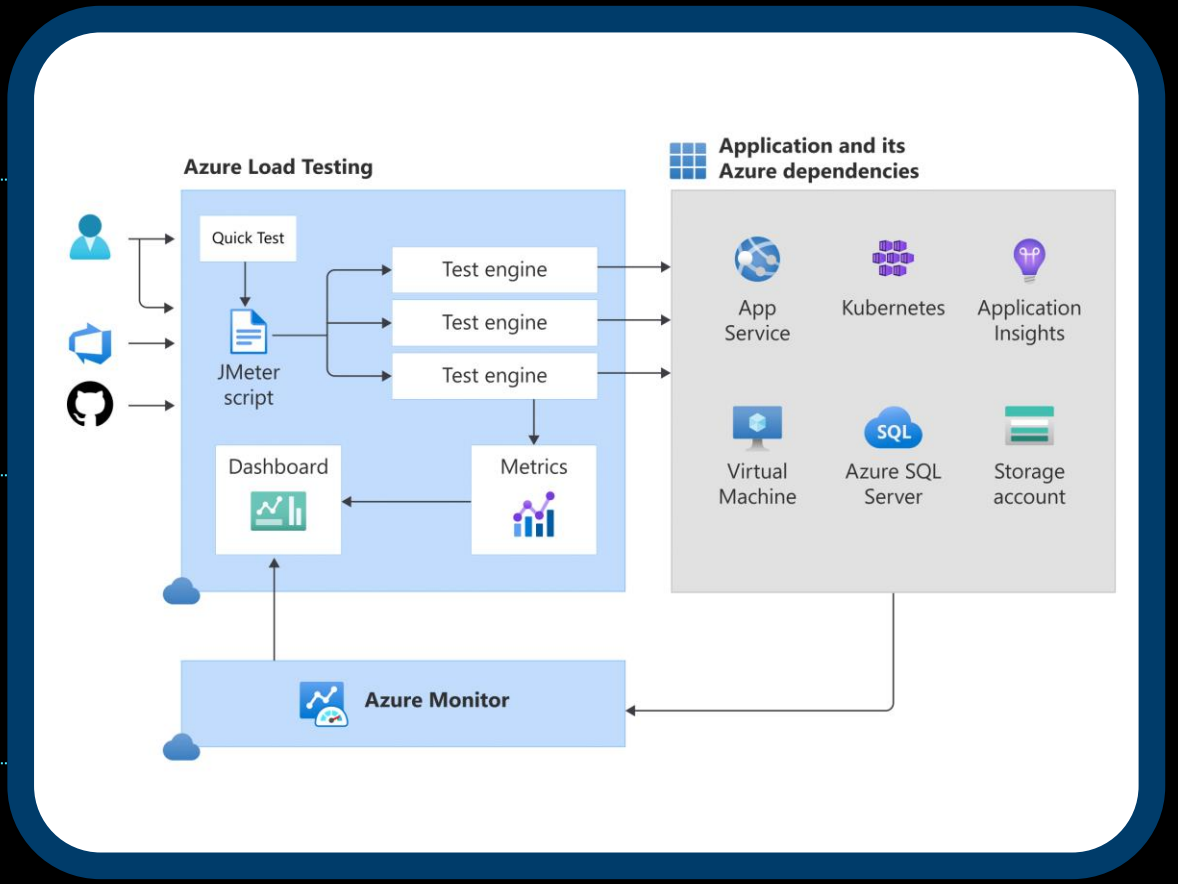
- Managing your own infrastructure
- Scalability
- Many teams need to be coordinated
- Happens on project milestones and not enough times

Azure Load Testing at a glance

Eliminate infrastructure needs with a fully managed service

Run existing test scripts with high-fidelity JMeter support

Integrated in both GitHub Actions and Azure Pipelines



Demo test if your code can handle the load

Build load testing into DevOps workflows

Shift left

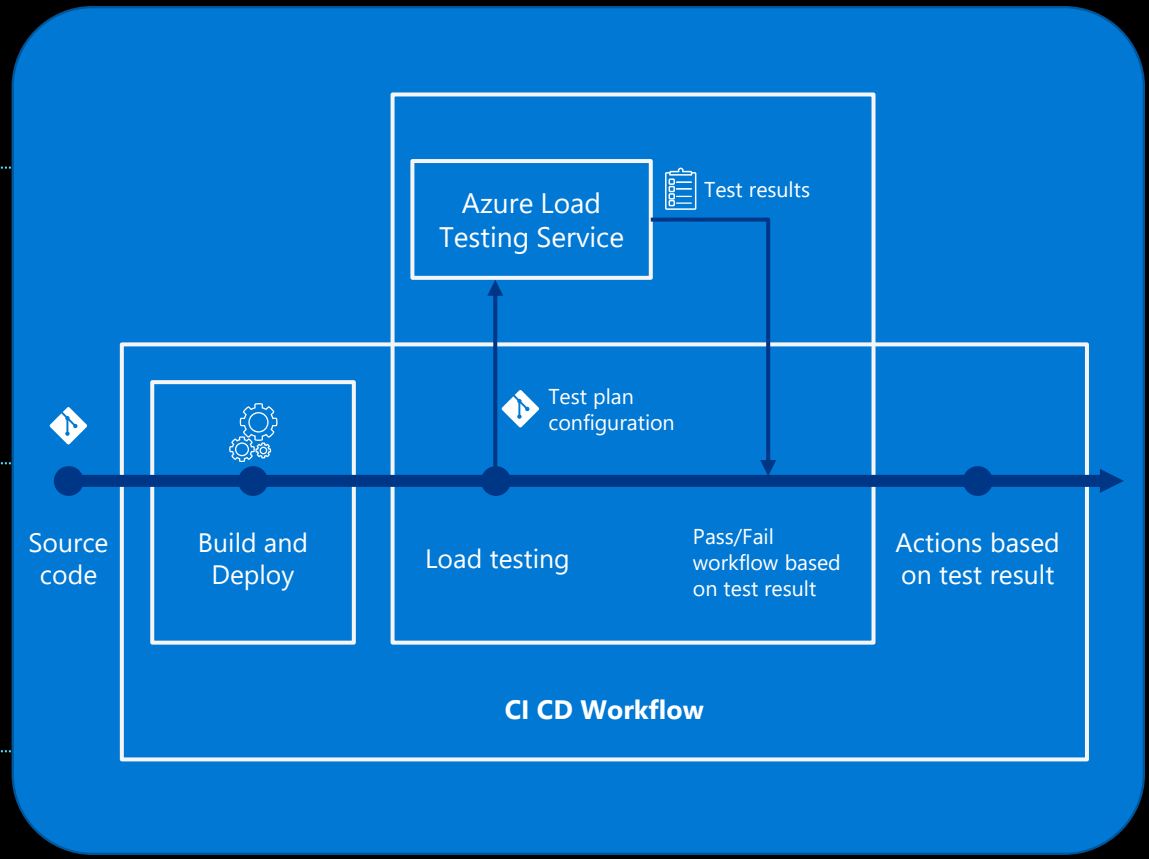
Shift load testing left to continually improve performance

Operate confidently

Establish a performance baseline and set clear pass/fail criteria

Innovate faster

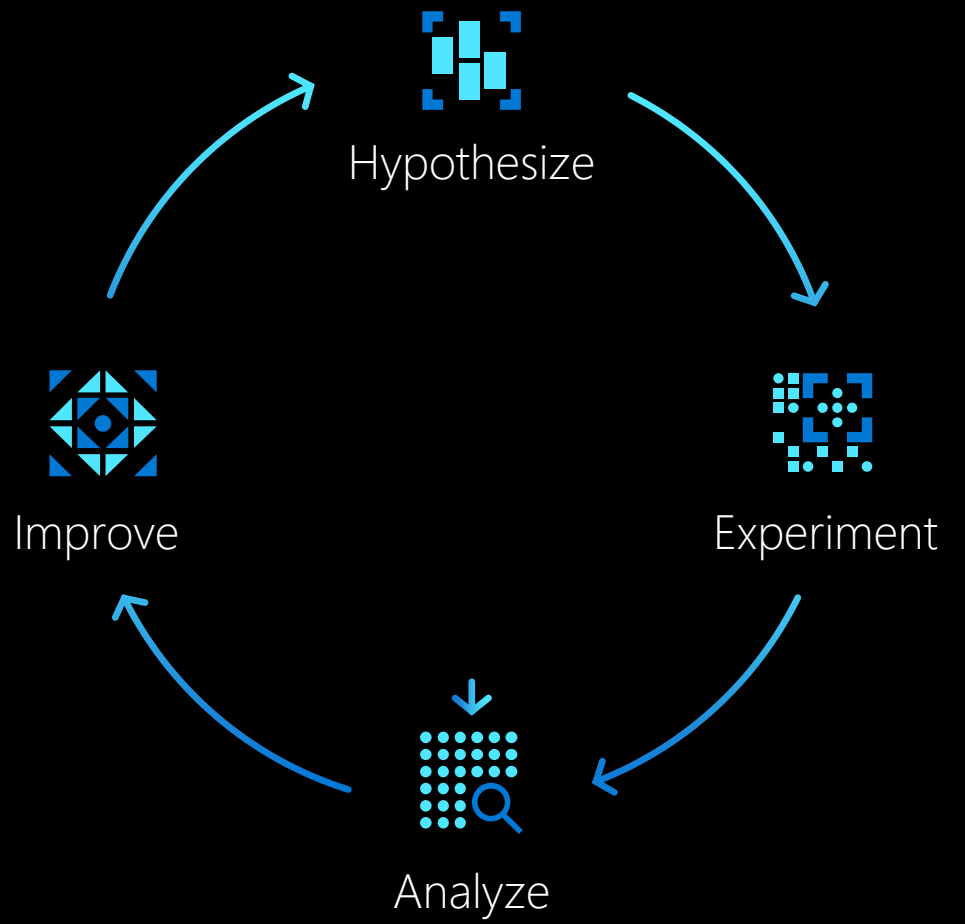
Automatically export test results to GitHub or Azure DevOps with full context



Demo integrate your load test in the flow

How can I improve resilience further?

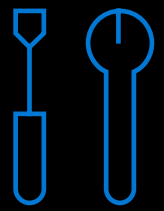
What is Chaos Engineering?



Resilience & Quality

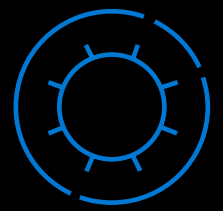
The capability of a system to handle and recover from disruptions

Service outages impact availability



Unavailability causes business impact – upset customers but also financial, legal, life-or-death consequences

Outages also steal developer time from working on improvements + new features



Quality practices need to be built-in to the entire service development and operation lifecycle

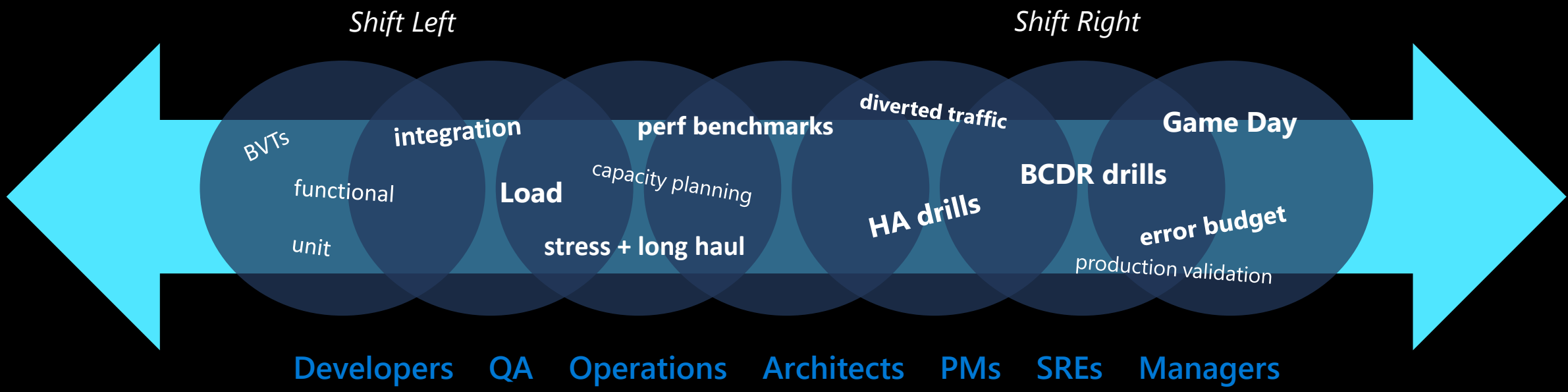
Resilience in the cloud

Cloud applications must be designed to handle failure.

When running applications in the cloud, resilience becomes a shared responsibility.

Azure needs tooling that helps its own services and customers using those services to build resilience.

Chaos Engineering Scenarios



Measure, understand, improve, and maintain product resilience

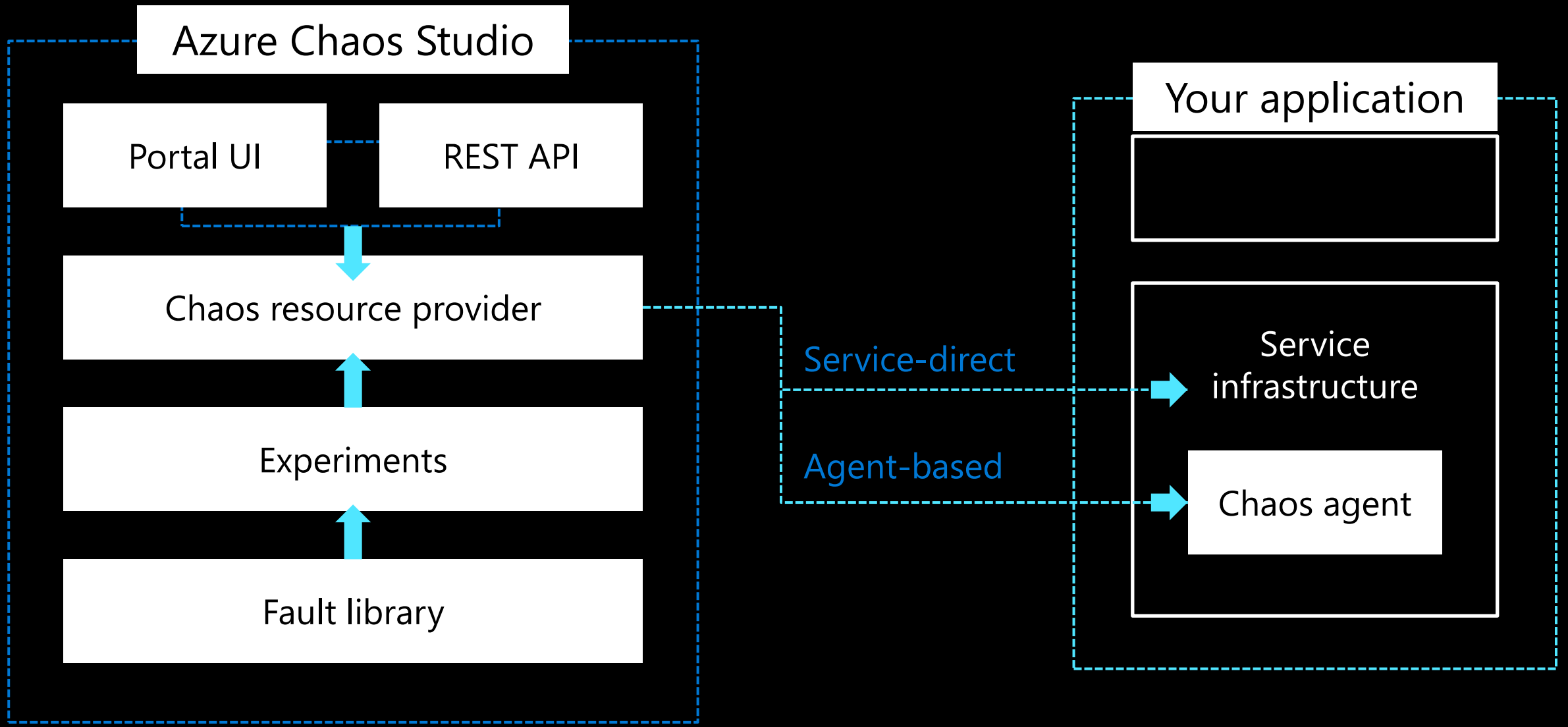
Fully managed service for building resilience into your Azure services

Deeply integrated into Azure, including Azure Resource Manager, Azure Monitor, and Azure Active Directory.

Expanding library of faults for common Azure service issues

Simulation of real-world scenarios using orchestrated parallel and sequential fault injection

Stop and roll back experiments to safeguard fault injection from causing outages


Azure Chaos Studio
Portal UI
REST API
Chaos resource provider
Experiments
Fault library
Service-direct
Agent-based
Your application
Service infrastructure
Chaos agent

Thank you!