
AWS
re:Invent

BLD202

Building technology standards at Amazon scale

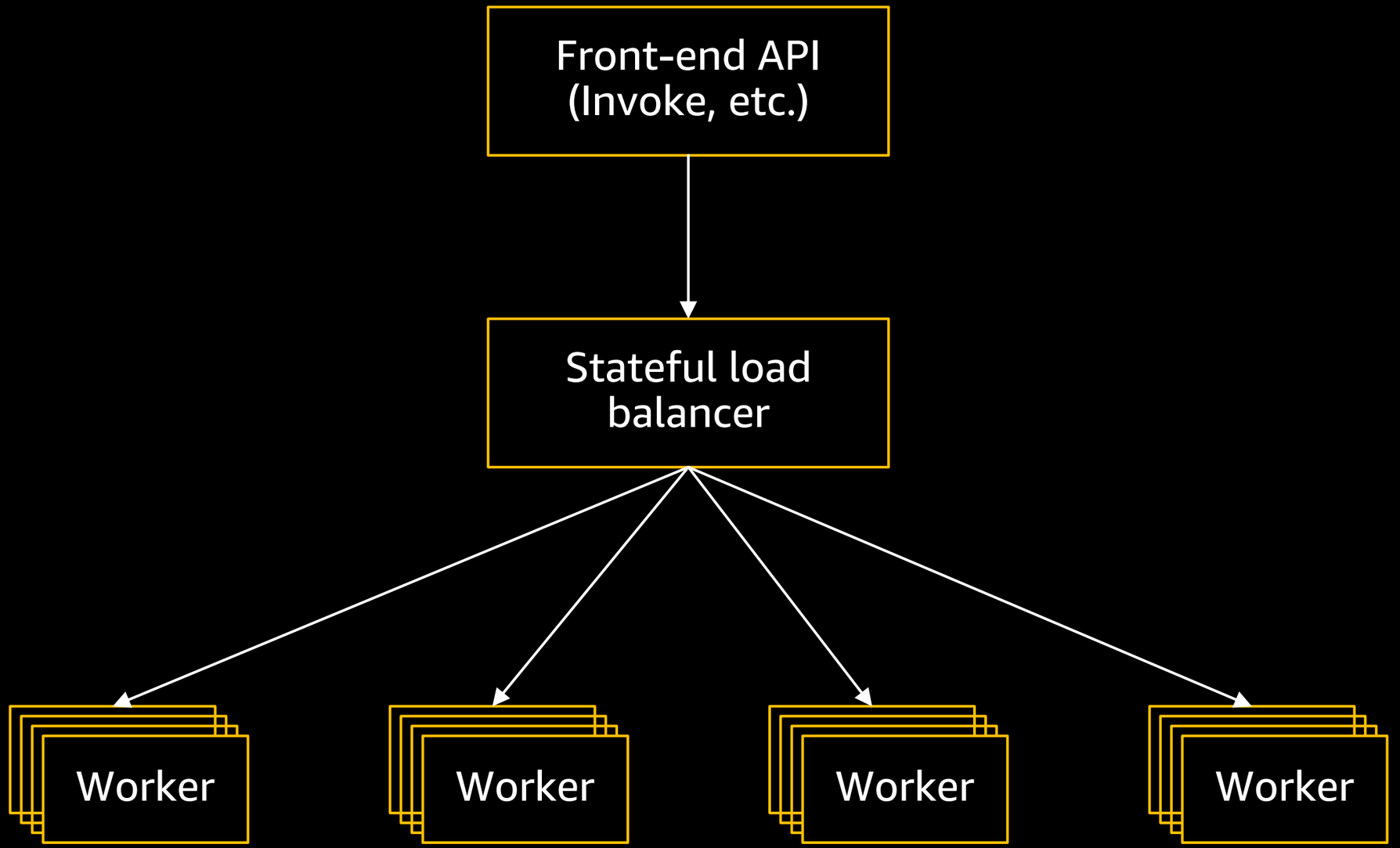
Marc Brooker

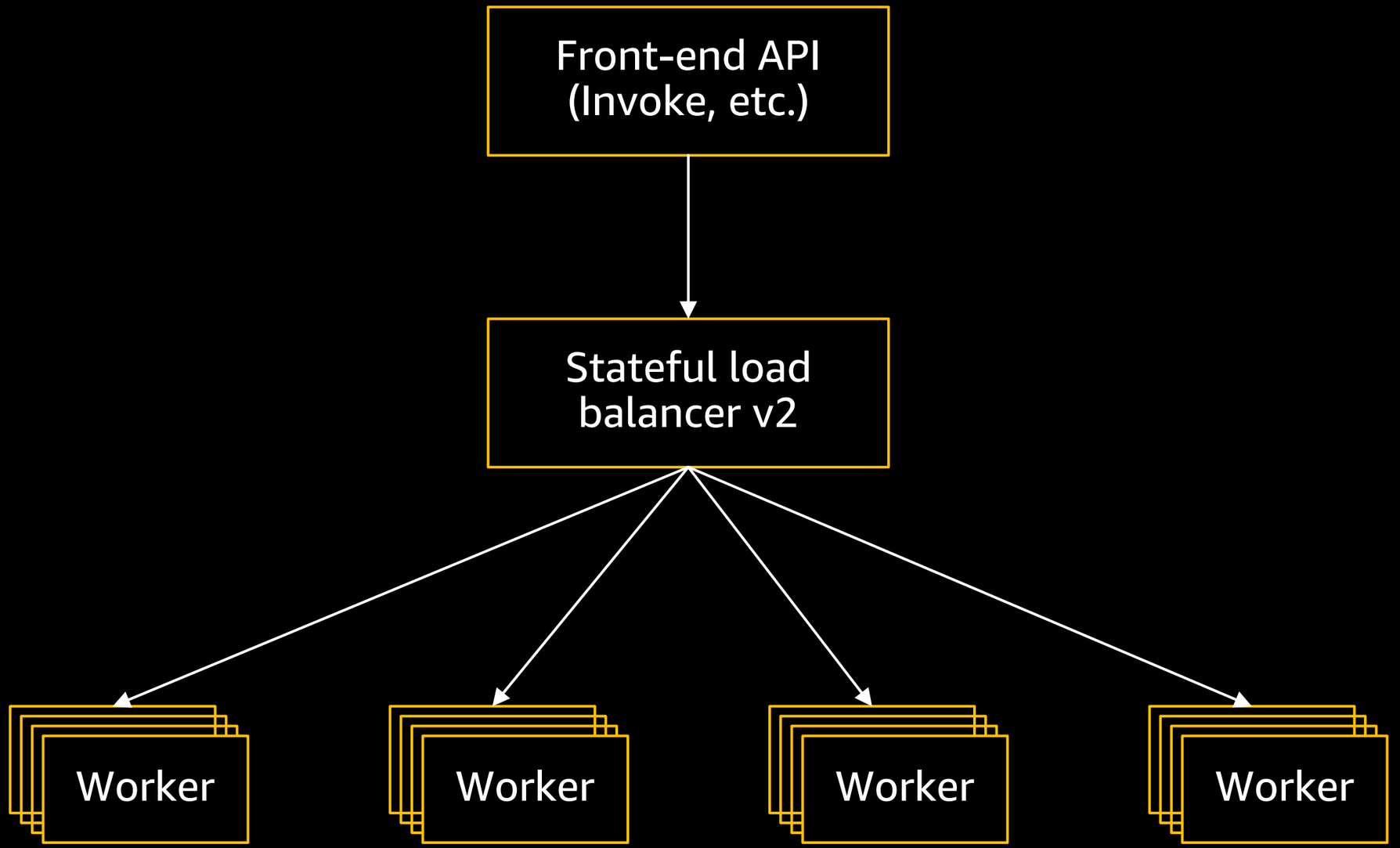
Senior Principal Engineer

AWS

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.


Front-end API
(Invoke, etc.)
Stateful load
balancer
Worker
Worker
Worker
Worker


Front-end API
(Invoke, etc.)
Stateful load
balancer v2
Worker
Worker
Worker
Worker

Rust at AWS



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Why?

- Great performance
  - Especially at high percentiles
- Memory safety
- Multithreading
- Static analysis

Why not?

- Higher risk
  - Team not as familiar with technology
  - Relatively new
  - Small external community

Use what you have

vs.

Try new things

Innovation with guard rails

(both technical and cultural)

Cultural guard rails: Tenets

Think long-term

Software lasts a long time. The costs of operations and maintenance outweigh the initial cost of development. We think long-term about our tool and language choices, and we must be willing to contribute to improvements in whatever tools we use.

Technical guard rails

Isolate with APIs

Services should offer APIs and contracts that do not leak language choices. Especially avoid baking language-specific choices into data at rest.

Management instincts →

Engineer instincts →

Use what you have

vs.

Try new things

Tenets in tension

Fewer is better

Supporting each individual language adds cost and risk

Specialize

Picking the right language for the right job has considerable benefits for our business and customers

Standards?



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

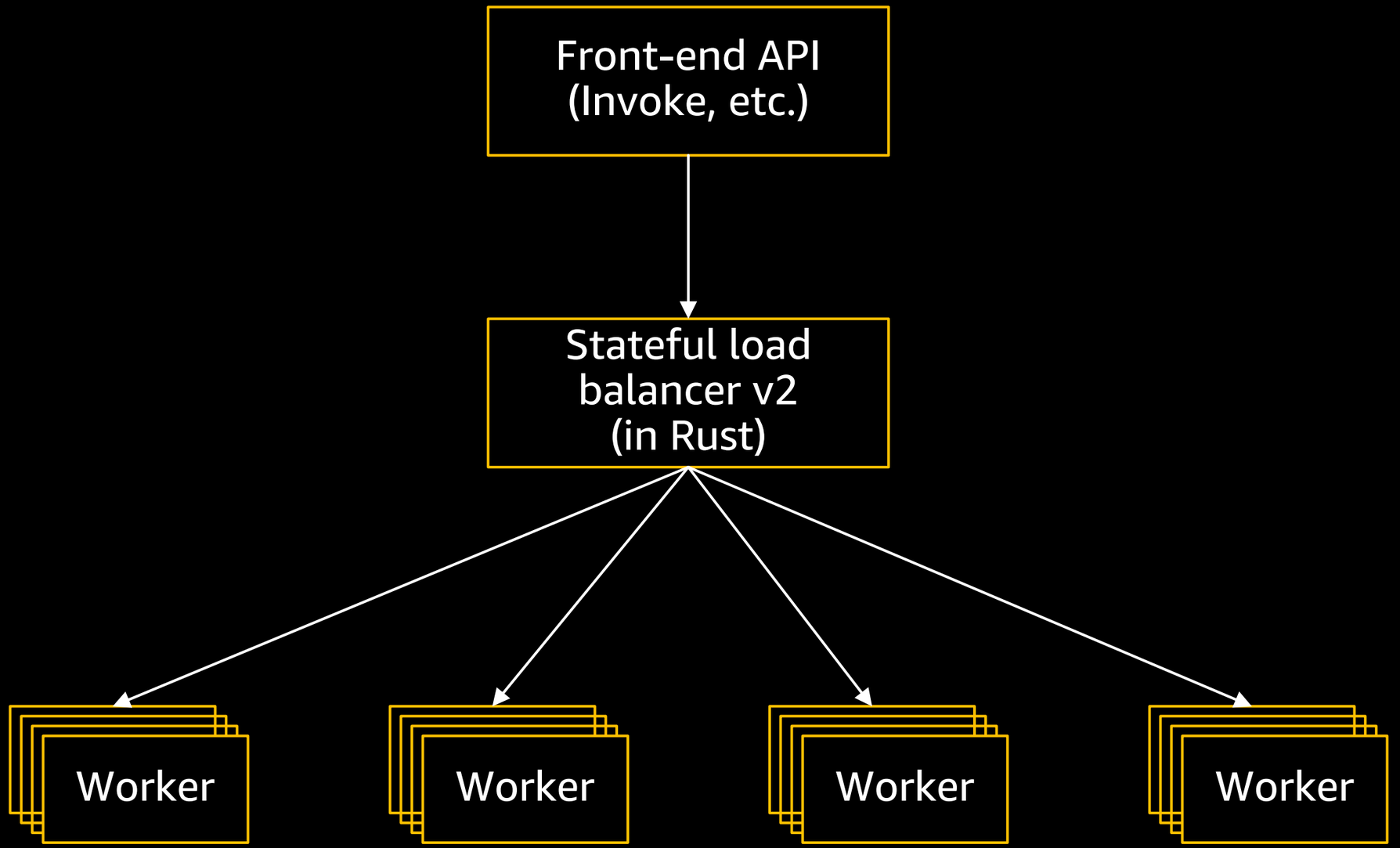
Standards may be necessary

Incentives

Ownership

Understanding

Your teams can’t make good decisions unless they understand the business and technology


Front-end API
(Invoke, etc.)
Stateful load
balancer v2
(in Rust)
Worker
Worker
Worker
Worker

Rust at AWS



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Double down on successes

We’re building in Rust

Firecracker

https://firecracker-microvm.github.io/

BottleRocket

https://aws.amazon.com/bottlerocket/

And lots more to come

We’re investing directly in Rust

- Hiring engineers on compiler, core, and key projects
- Focusing on long-term
  - Health of the community
  - Technology innovation
- Applying formal methods to Rust code

Back to tenets

Think long-term

Software lasts a long time. The costs of operations and maintenance outweigh the initial cost of development. We think long-term about our tool and language choices, and we must be willing to fund improvements in whatever tools we use.

Thank you!

Marc Brooker

Senior Principal Engineer
AWS
@marcjbrooker

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Please complete the session survey

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.