Serverless

for data and AI

Avner Braverman

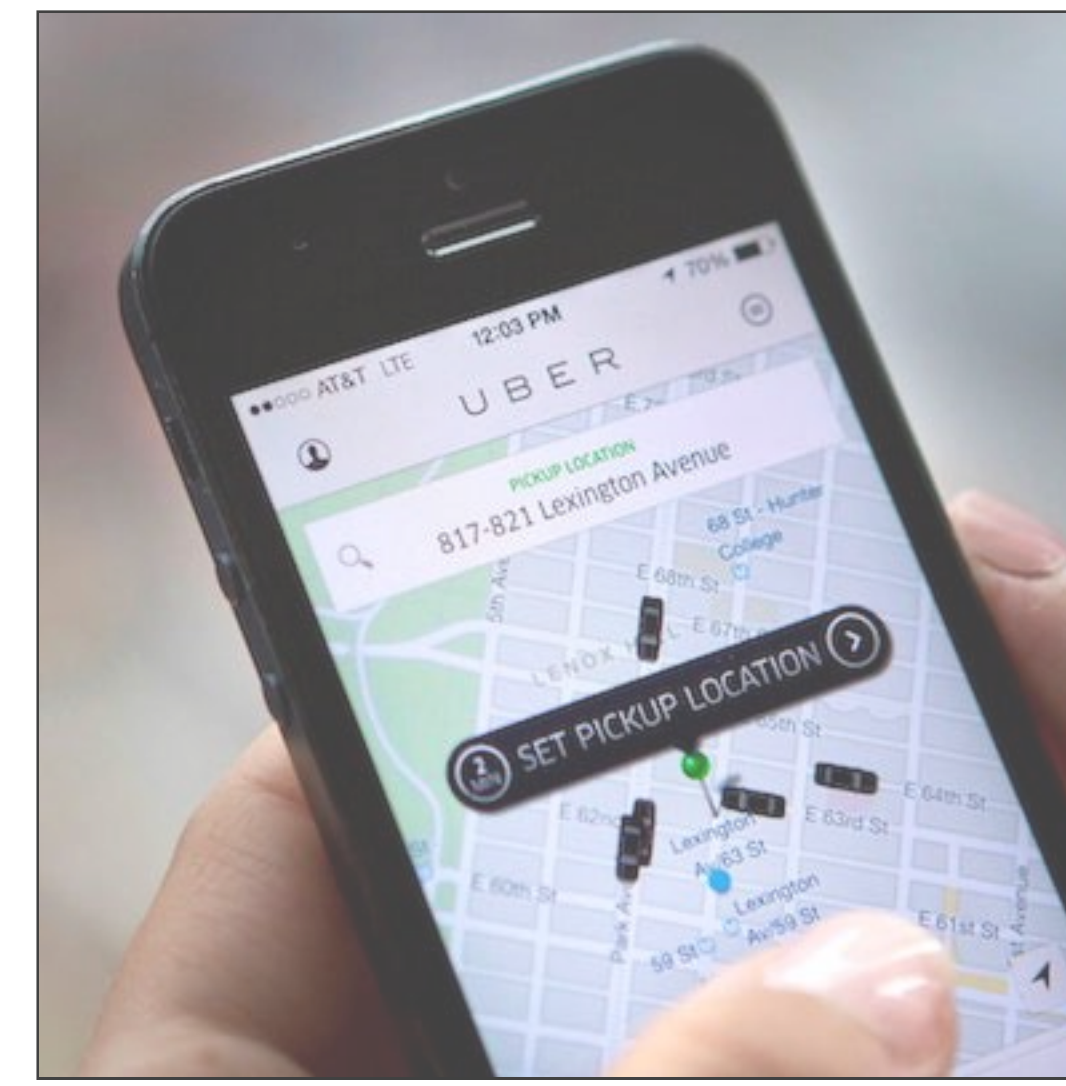
Binaris CEO

The future is Serverless

On-Prem

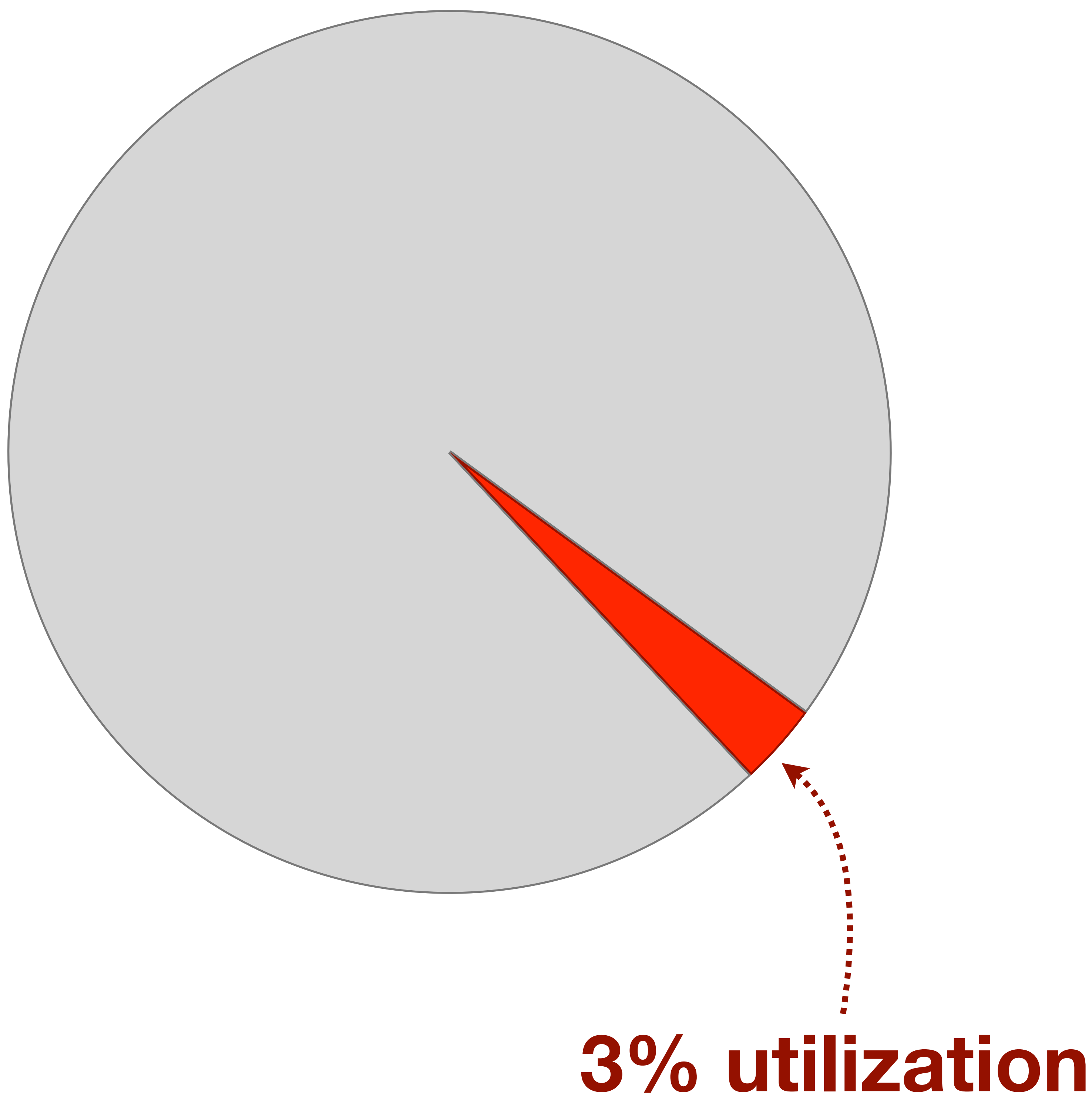
Cloud

Serverless

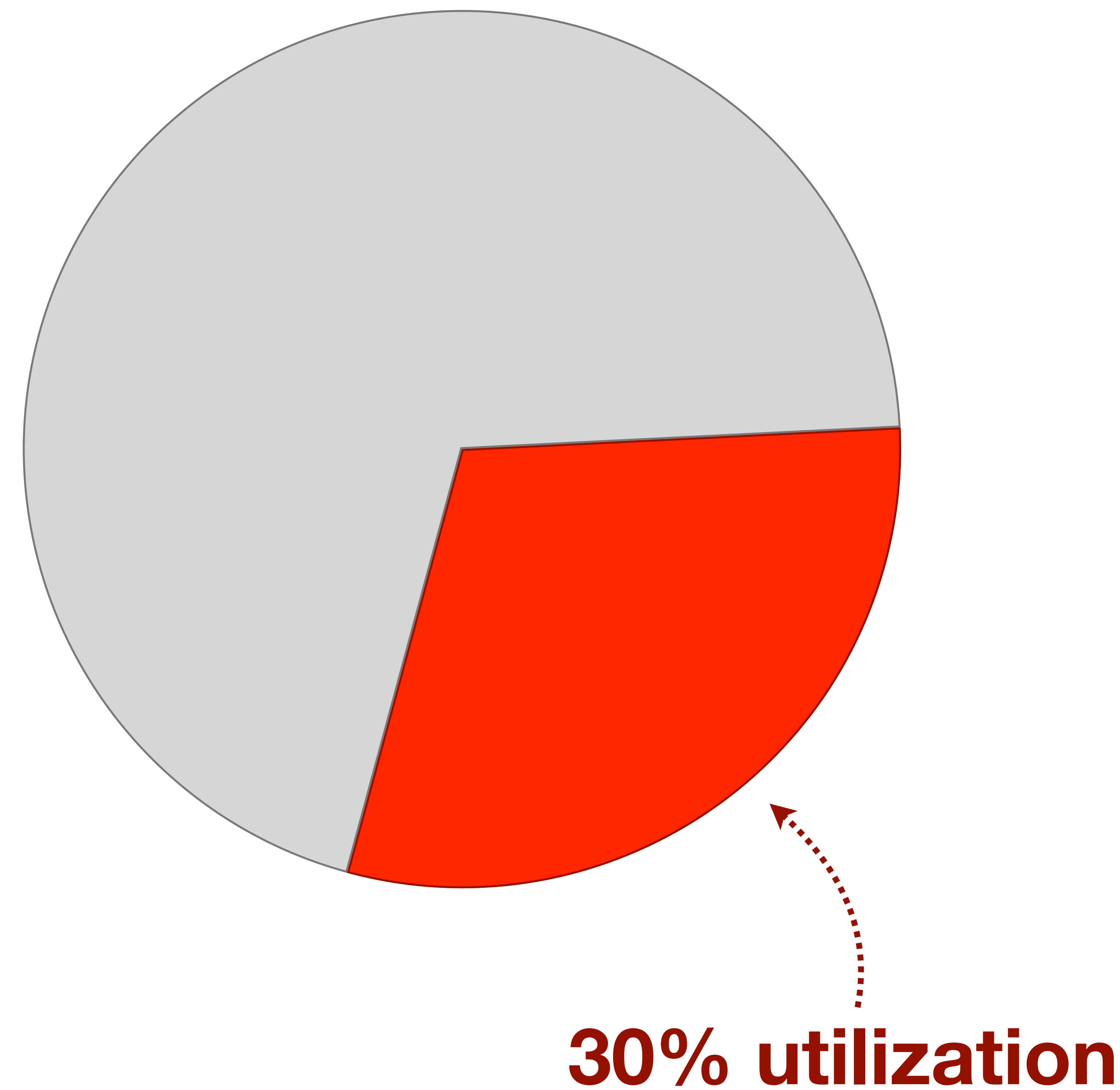


	Ride sharing	Serverless
Infrastructure	Someone else's car	Never manage servers
Cost	Pay per ride	Pay per CPU time

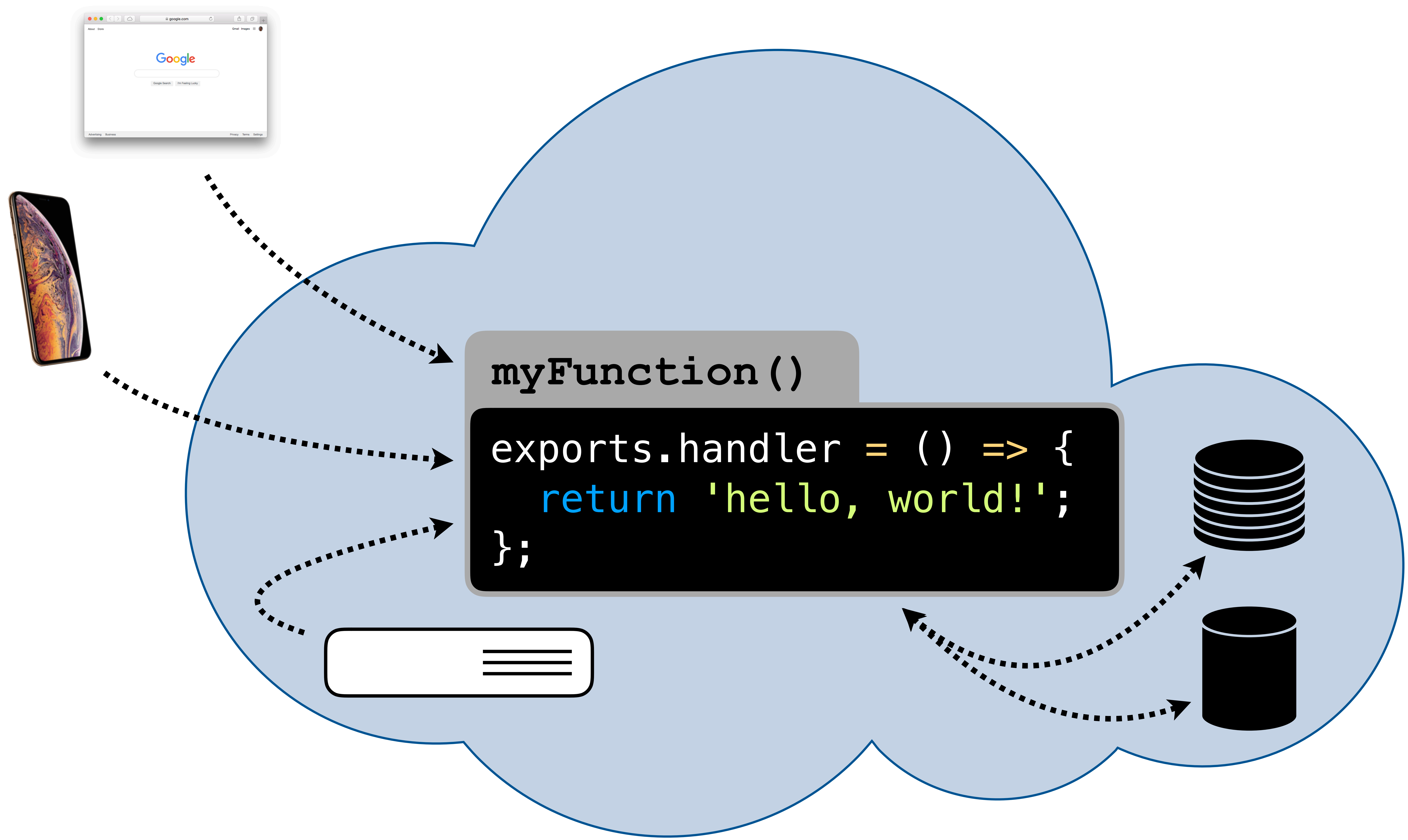
Legacy applications



Cloud-Native applications



	Ride sharing	Serverless
Infrastructure	Someone else's car	Never manage servers
Cost	Pay per ride	Pay per CPU time
Ease of use	Focus on your phone	Focus on business logic
Who can use?	No driver's license	No cloud expertise



Google
myFunction()
exports.handler = () => {
return 'hello, world!';
};

Demo

Key providers

AWS Lambda

Azure Functions

Binaris

Google Cloud Functions

IBM Cloud Functions

① Serverless ETL

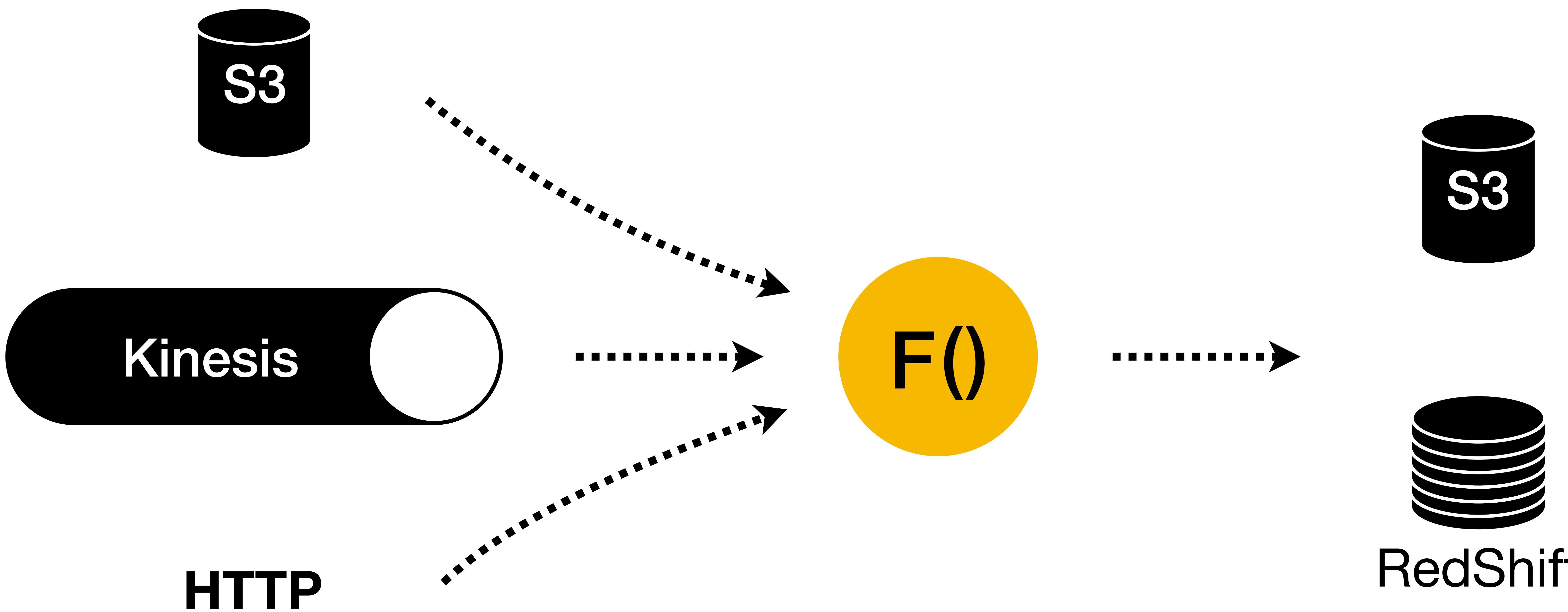
② Serverless MapReduce

③ Serverless streaming

④ Serverless training

⑤ Serverless inference

Serverless ETL



Serverless ETL

Upsides

Easy to build

Elastic

Usually cheaper to run

Challenges

Micro-batching

Run duration limits (15 min on AWS Lambda)

Memory limits (up to 3GB on AWS Lambda)

ETL accounts for 90% of Lambda CPU cycles

① Serverless ETL

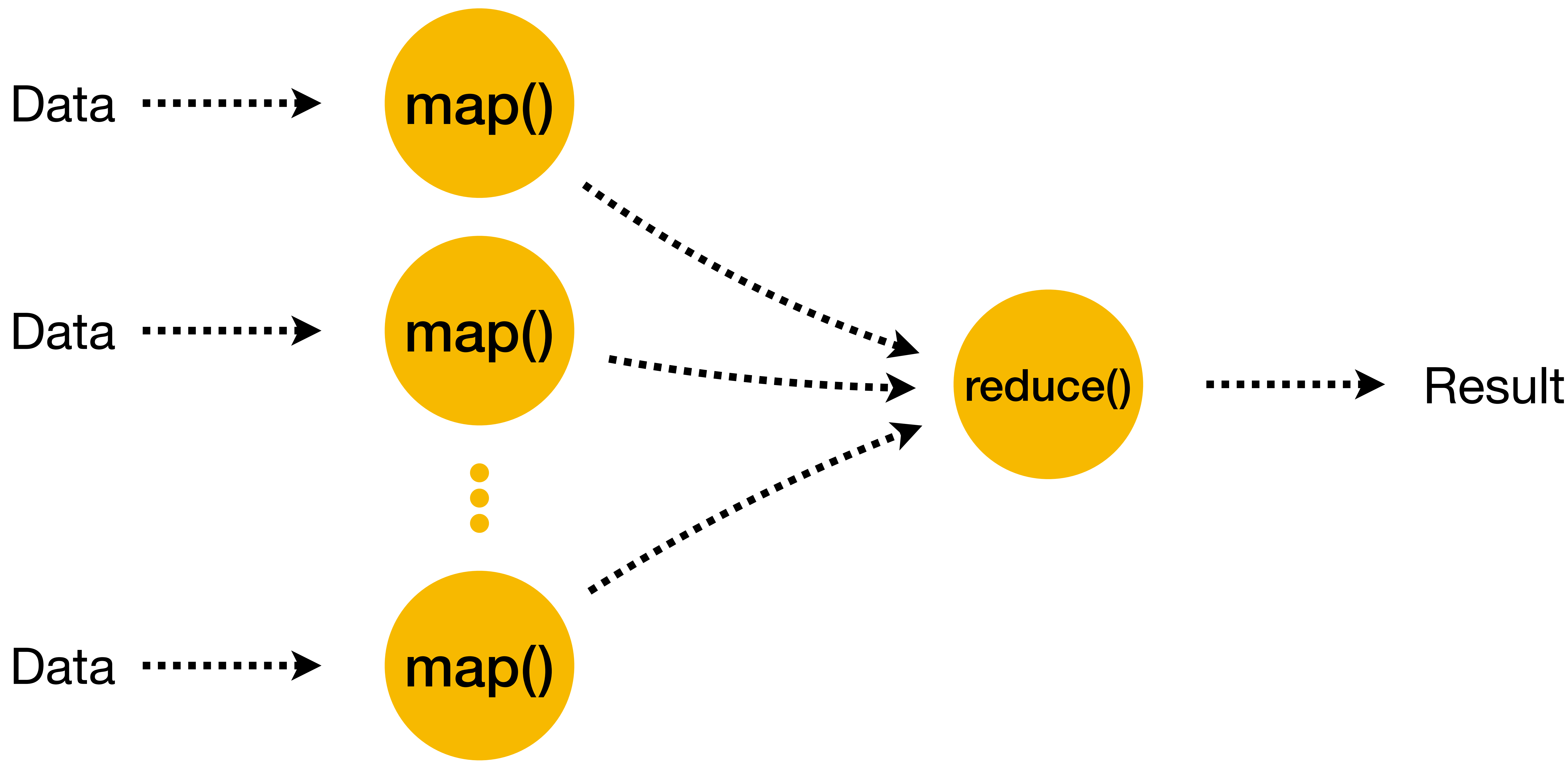
② Serverless MapReduce

③ Serverless streaming

④ Serverless training

⑤ Serverless inference

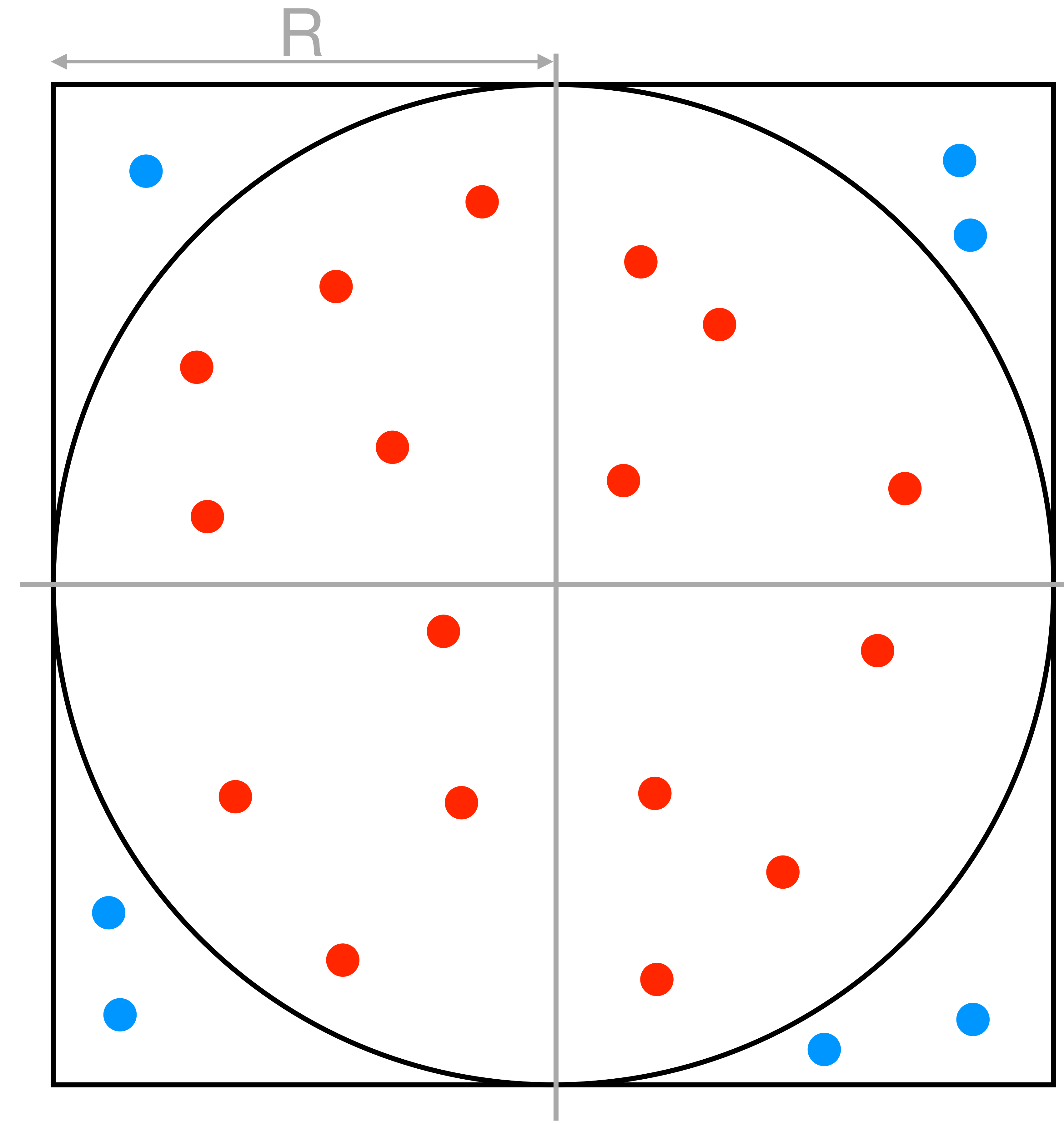
Serverless MapReduce



Serverless MapReduce

Calculating π using the Monte Carlo method:

1. Spread random points inside a square
2. Geometry tell us that
   Circle area ~ # red points ~ $\pi \cdot R^2$
   Square area ~ # points ~ $4 \cdot R^2$
3. Compute:
   $\pi = 4 * (\text{\# red points}) / (\text{\# points})$



Serverless MapReduce

```
function computePi(points) {
  let inside = 0;

  // repeat points times:
  for (let i = 0; i < points; i++) {

    // random point (use R = 1)
    const x = Math.random() * 2 - 1;
    const y = Math.random() * 2 - 1;

    // is it inside the circle?
    if (x * x + y * y < 1) {
      inside++;
    }
  }
  return inside;
  // use circle to square area ratios to compute π
  return 4 * inside / points;
}
```

More points == Better π

Serverless MapReduce

```
exports.computePiMapper = ({ points }) => {
  let inside = 0;

  // repeat points times:
  for (let i = 0; i < points; i++) {

    // random point (use R = 1)
    const x = Math.random() * 2 - 1;
    const y = Math.random() * 2 - 1;

    // is it inside the circle?
    if (x * x + y * y < 1) {
      inside++;
    }
  }

  return inside;
}
```

① Data

② Map

```
function mrController(...) {
  // invoke mappers
  // wait for completion
  // invoke reducer
  // return result
}
```

③ Reduce

```
exports.computePiReducer = ({ inputs, points }) => {
  return 4 * inputs.reduce((a, b) => a + b) / points;
};
```

Demo

Serverless MapReduce

Upsides

Easy to build

Elastic

Scale at your fingertips

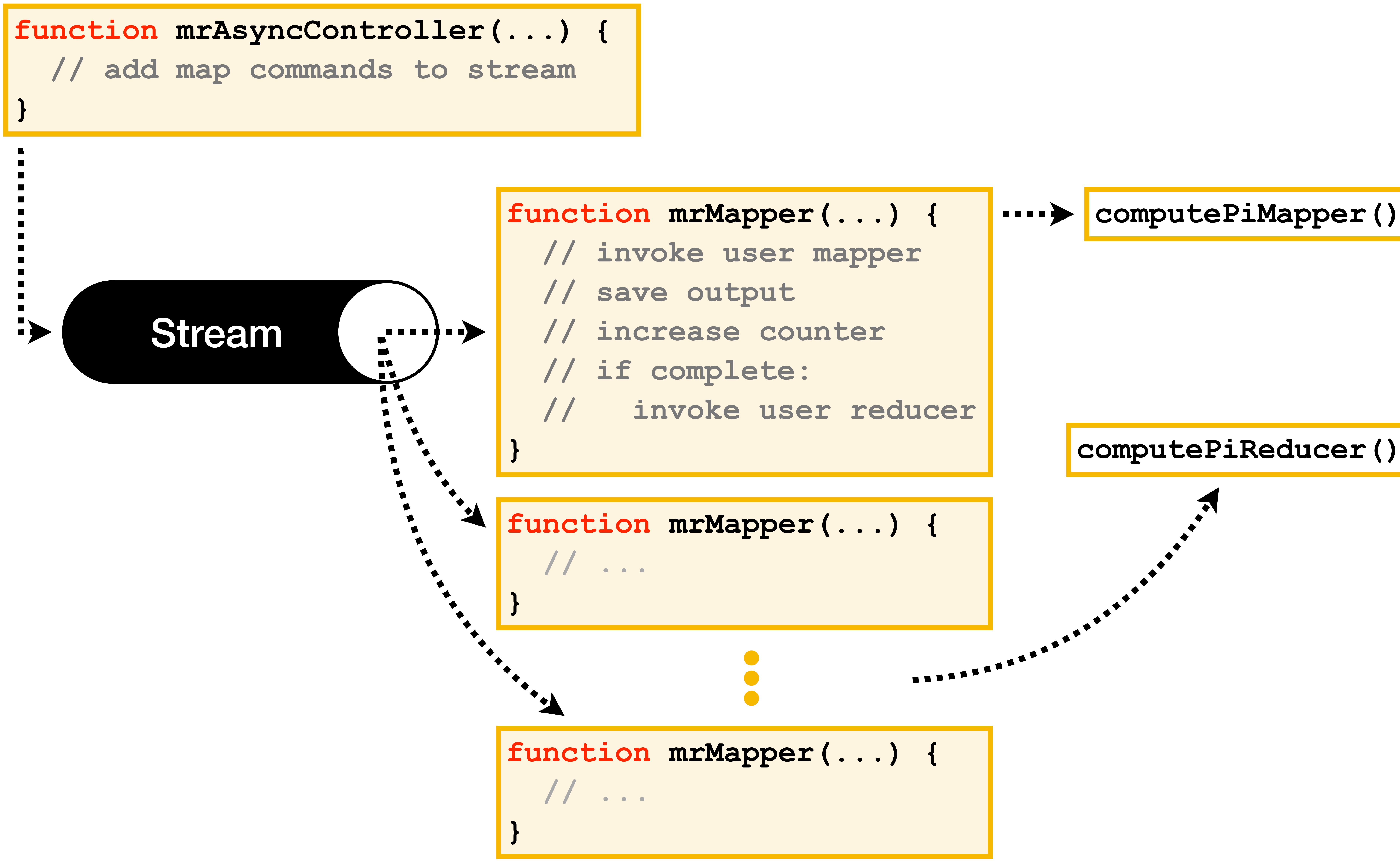
Challenges

Failures?

Retries?

Laggards?

Serverless MapReduce



Serverless MapReduce

github.com/binaris/functions-examples

Serverless MapReduce

Upside

Easy to build (with framework)

Elastic

Scale to your fingertips

Challenges

Need cache (e.g. Redis) to hold state

Need storage (e.g. S3) to handle data

① Serverless ETL

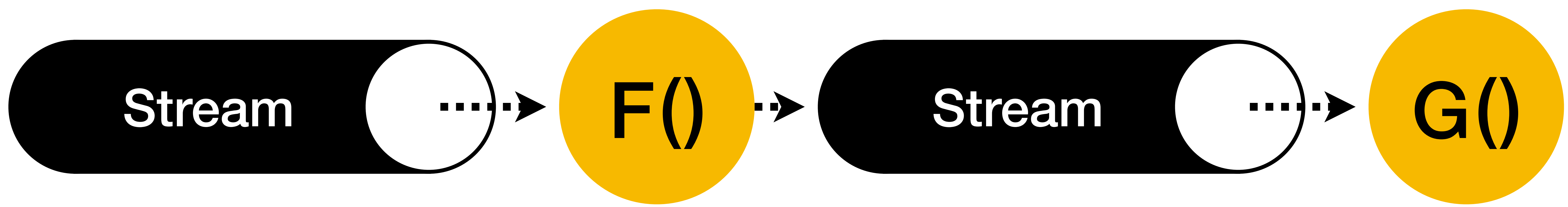
② Serverless MapReduce

③ Serverless streaming

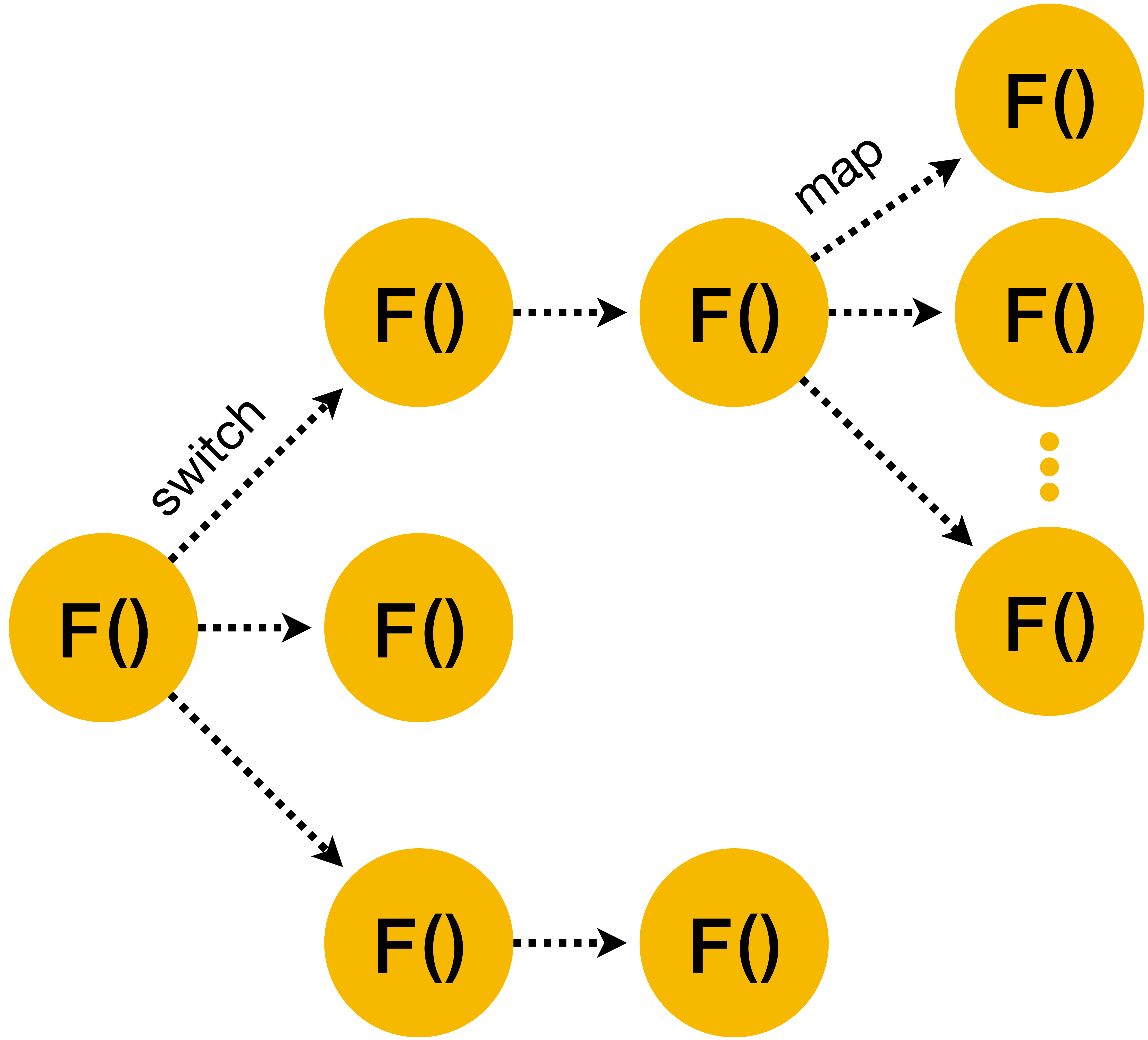
④ Serverless training

⑤ Serverless inference

Serverless streaming



Serverless streaming



Serverless streaming

Upsides

Easy to build

Elastic

Scale to your fingertips

Real-time

Challenges

Need a smarter framework

Shameless plug for Binaris

① Serverless ETL

② Serverless MapReduce

③ Serverless streaming

④ Serverless training

⑤ Serverless inference

Serverless training

Today, serverless is not a good fit for training 😕

But, if

① Your data is small (few GB)

② You need hyperparameter optimization

Then you can leverage serverless scale!

① Serverless ETL

② Serverless MapReduce

③ Serverless streaming

④ Serverless training

⑤ Serverless inference

Serverless inference

k-means

(Yeah, we know this is not really inference, but same computational pattern)

Demo

Serverless inference

Upsides

Easy to build

Use standard tools (py, sklearn, TF)

Elastic

Scale to your fingertips

Real-time (no more plugs)

Challenges

No GPU

Might be limited CPU/memory for some TF models

① Serverless ETL

② Serverless MapReduce

③ Serverless streaming

④ Serverless training

⑤ Serverless inference

Serverless
for data and AI

Serverless is easy to use

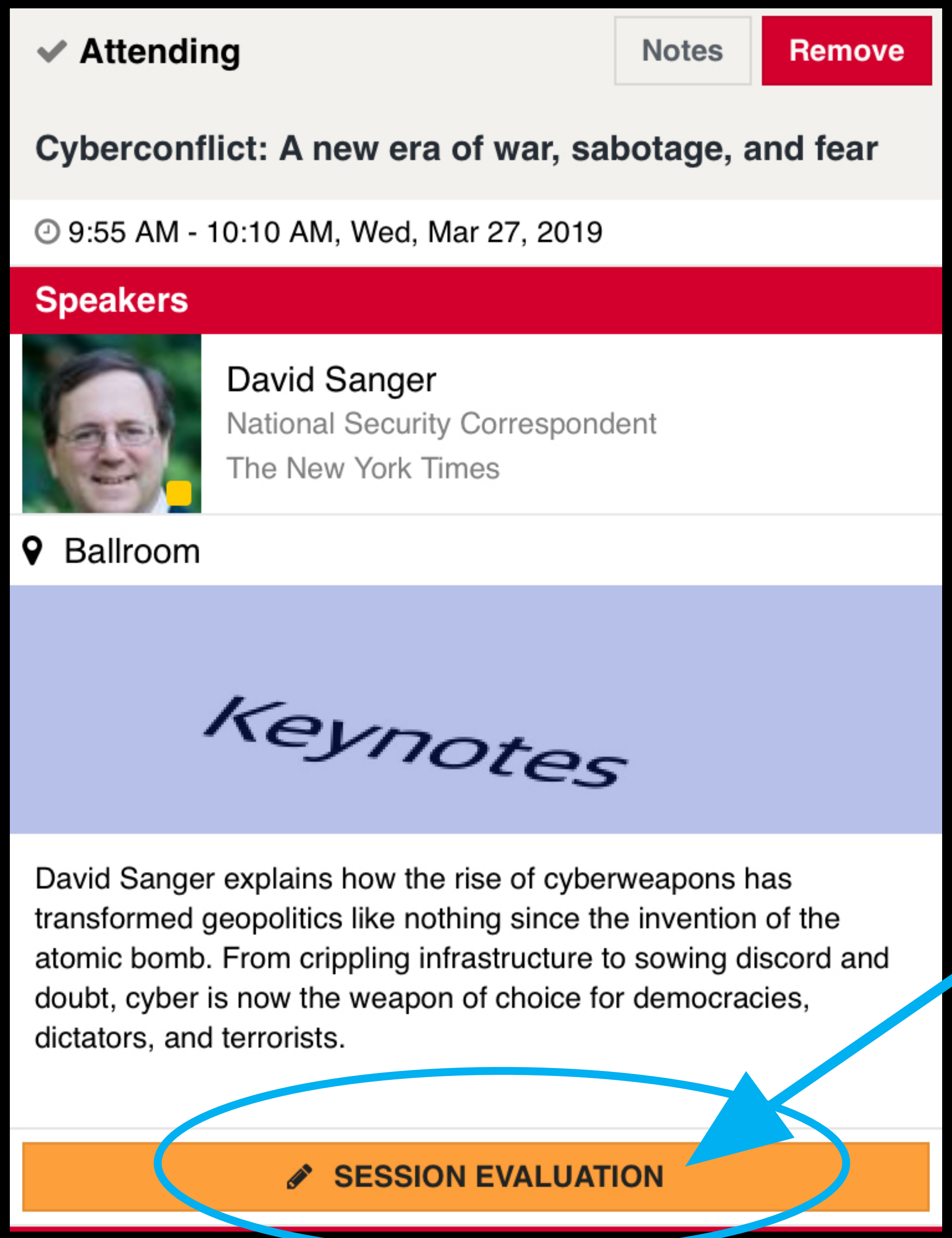
Cosmic ✨ scale without the rocket 🚀 science

Pay as you go

avner@binaris.com @avnerbraverman

Rate today's session

Session page on conference website



O'Reilly Events App