# An agile pathway towards carbon-aware clouds

**Pratyush Patel**, Theo Gregersen, Tom Anderson

UNIVERSITY *of* WASHINGTON

**Disclaimer: an *opinionated* talk with systems implications — let's discuss!**

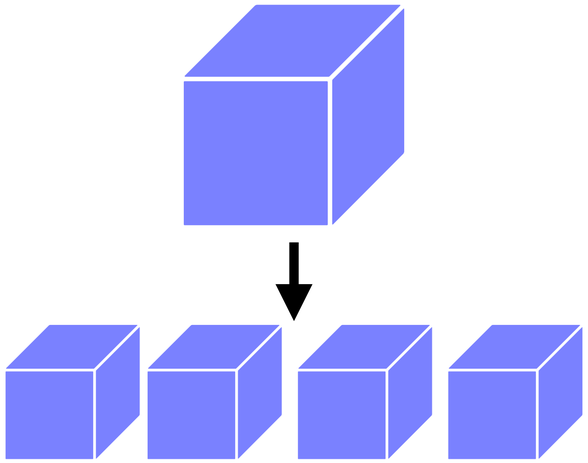
# How should cloud providers expose carbon awareness to users?

# How should cloud providers expose carbon awareness to users?

**Cloud users are**
***massive organizations***

# How should cloud providers expose carbon awareness to users?

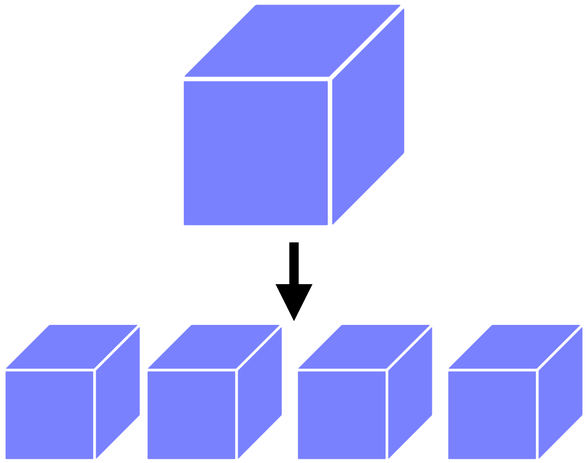

**Cloud users are**
***massive organizations***



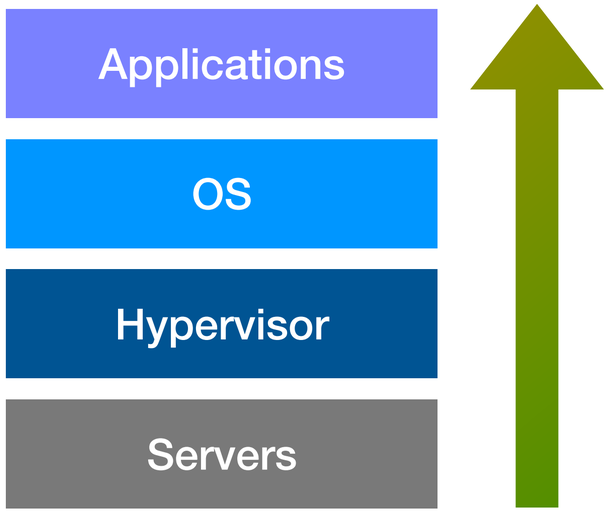
**Applications span**
***thousands of microservices***

# How should cloud providers expose carbon awareness to users?


**Cloud users are**
***massive organizations***



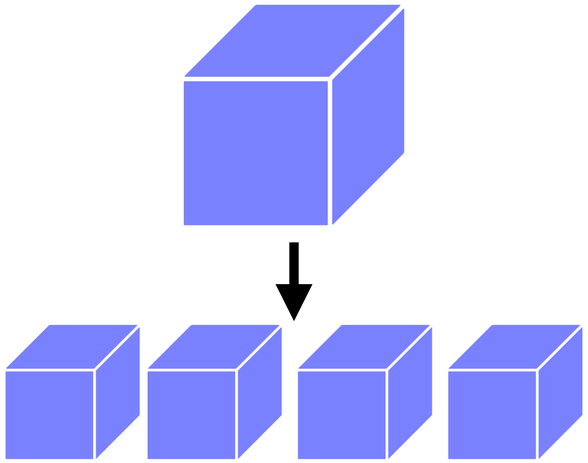
**Applications span**
***thousands of microservices***



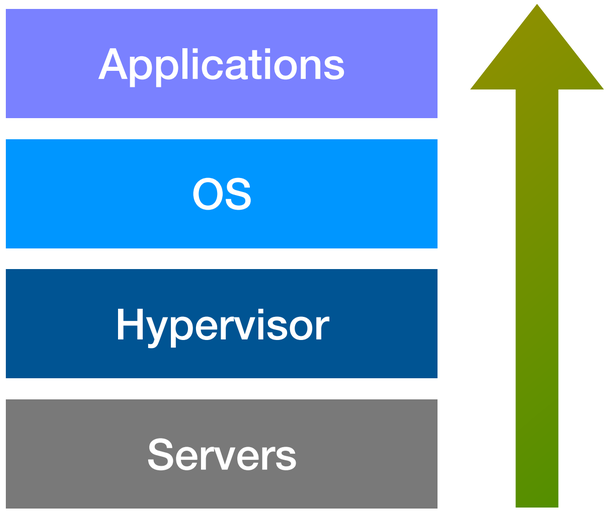

**Carbon awareness must**
***percolate the stack***

# How should cloud providers expose carbon awareness to users?


**Cloud users are**
***massive organizations***



**Applications span**
***thousands of microservices***

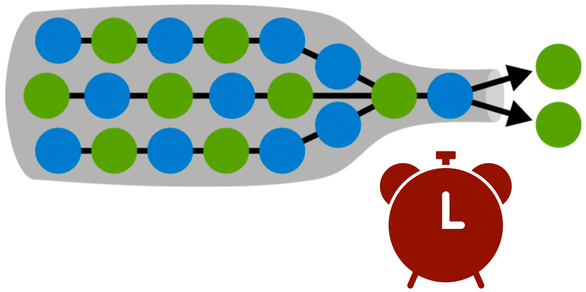



**Carbon awareness must**
***percolate the stack***

**Clearly, a very challenging problem!**

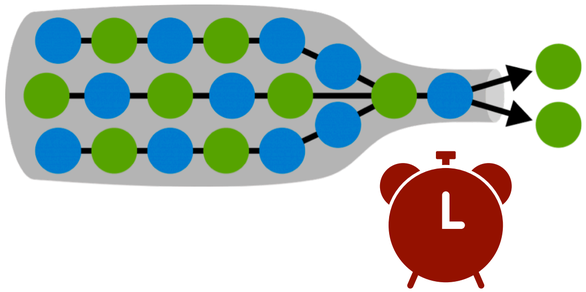
# Carbon APIs need time, effort, and care to deploy

# Carbon APIs need time, effort, and care to deploy

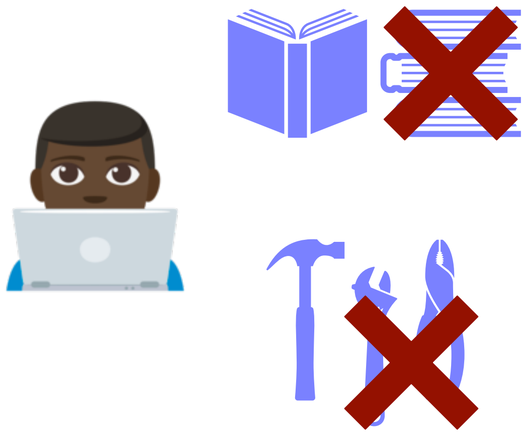


**Developer time is a *business bottleneck***

# Carbon APIs need time, effort, and care to deploy

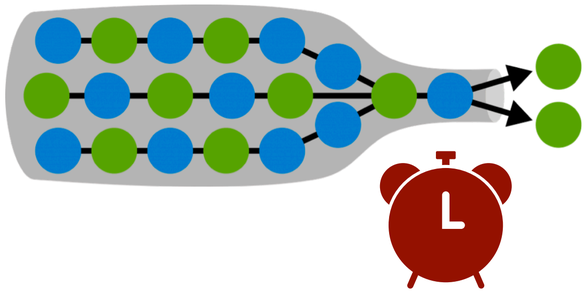


**Developer time is a *business bottleneck***

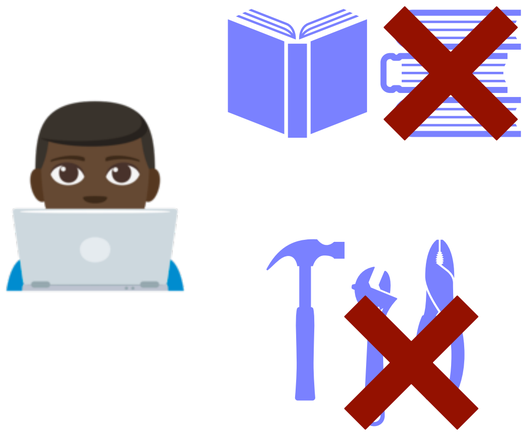


***Minimal training and tooling* available**

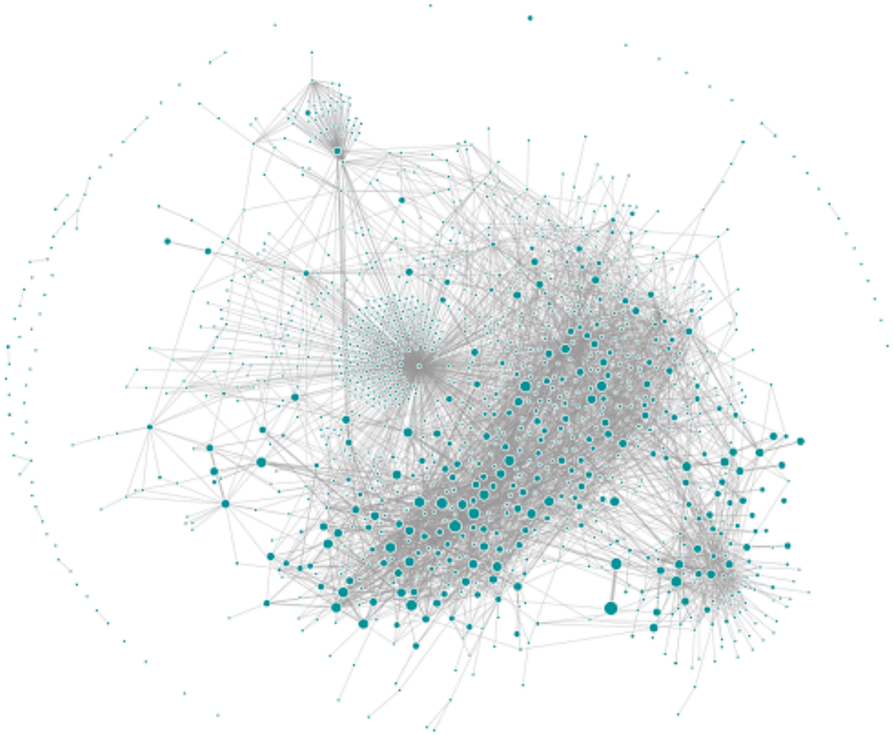
# Carbon APIs need time, effort, and care to deploy



**Developer time is a *business bottleneck***



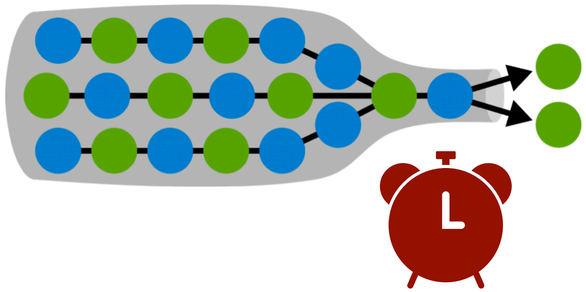
***Minimal training and tooling* available**



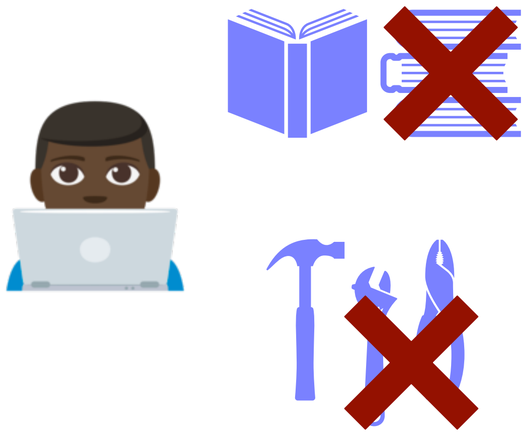
***Rewriting apps* may be *infeasible***

Uber's ~2,200 microservices visualized by Jaeger (2018)

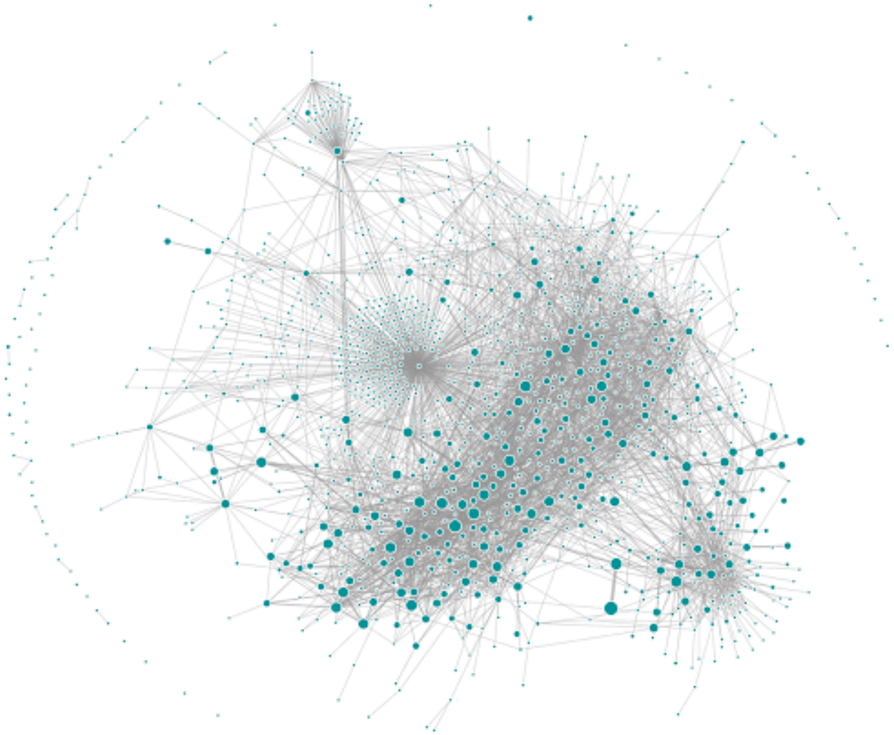
# Carbon APIs need time, effort, and care to deploy



**Developer time is a *business bottleneck***



***Minimal training and tooling* available**



***Rewriting apps* may be *infeasible***

**Who can make an impactful and timely difference? How can we help them?**

Uber's ~2,200 microservices visualized by Jaeger (2018)

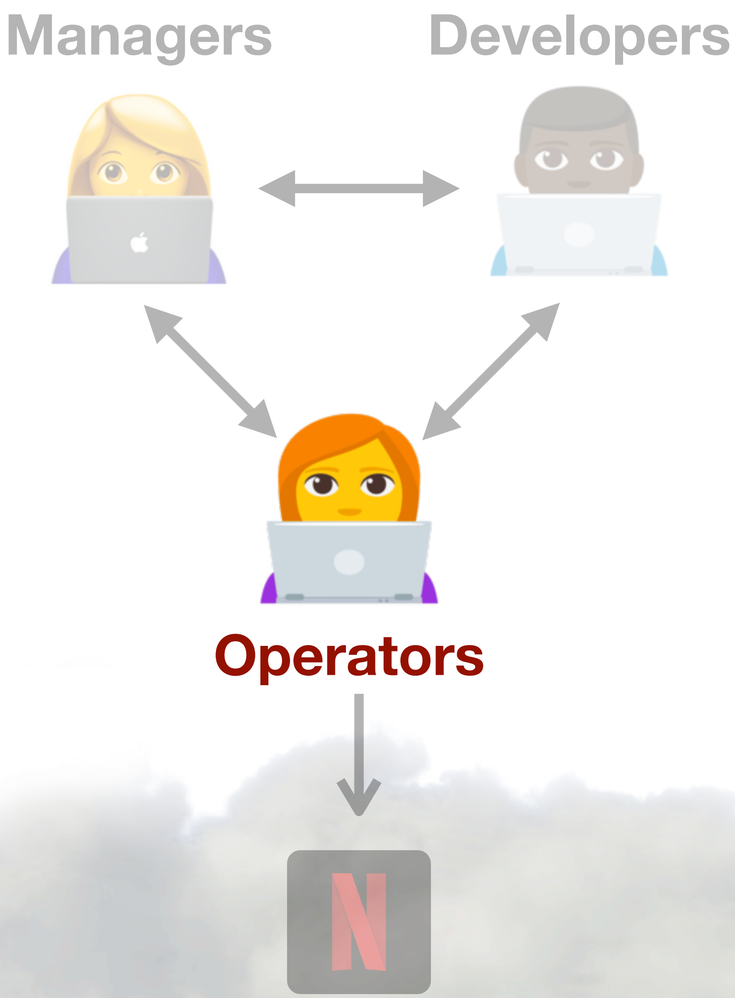
# A closer look at large-scale application teams


*simplified model

# A closer look at large-scale application teams

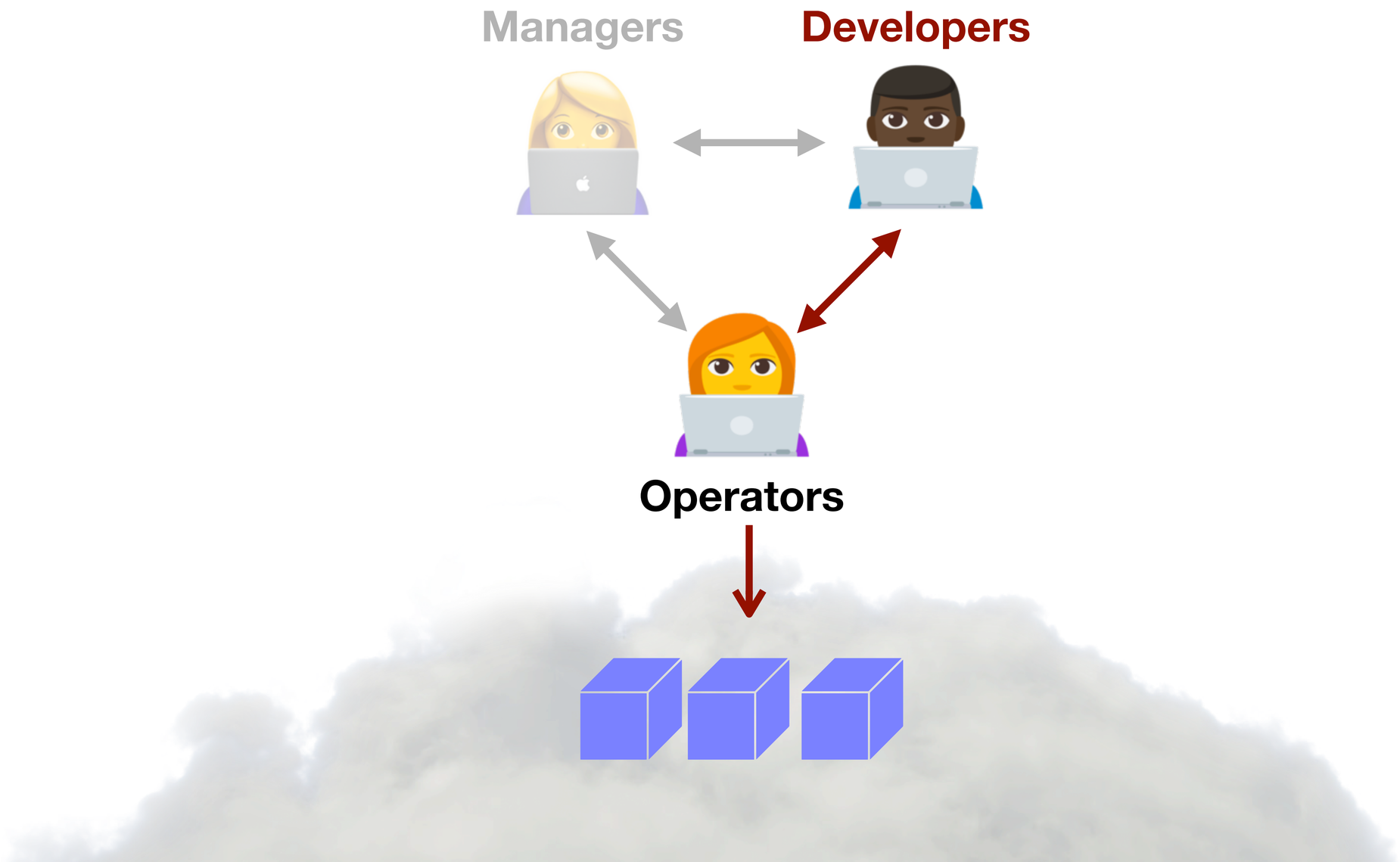

*simplified model

# Every large-scale application team has an operations team



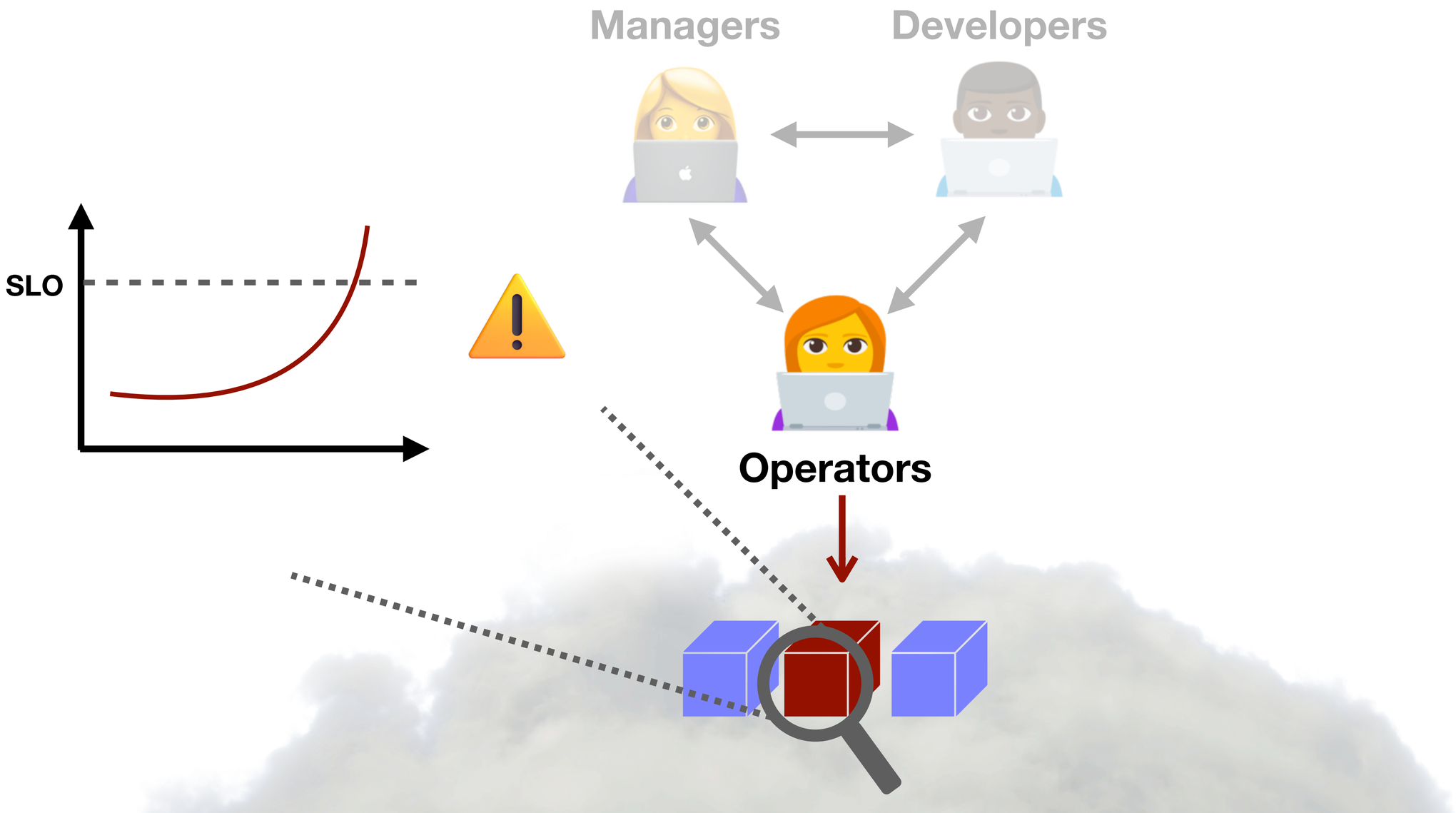

*simplified model

# Operators work with developers to deploy applications on the cloud



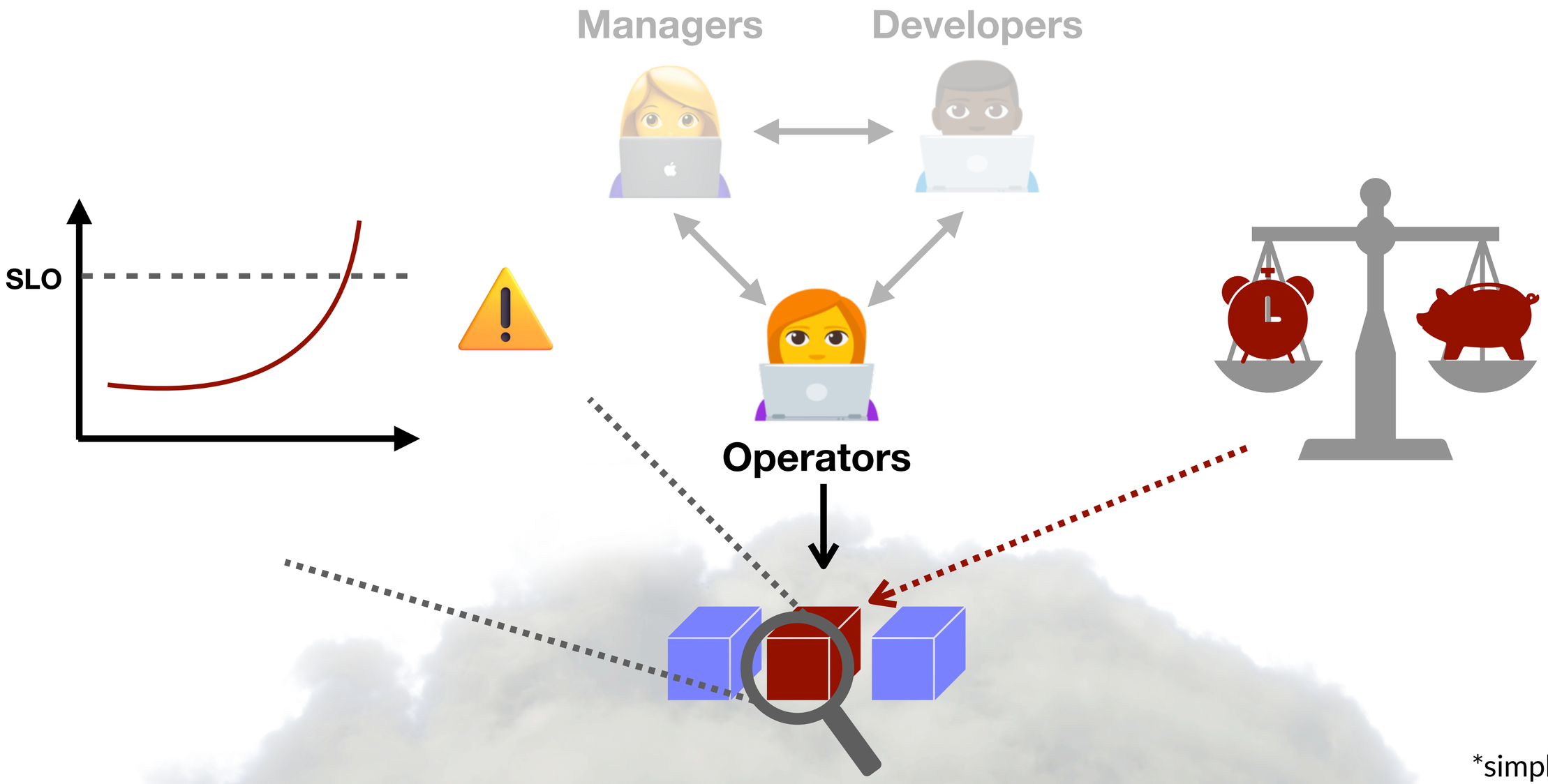

*simplified model

# Operators monitor deployments



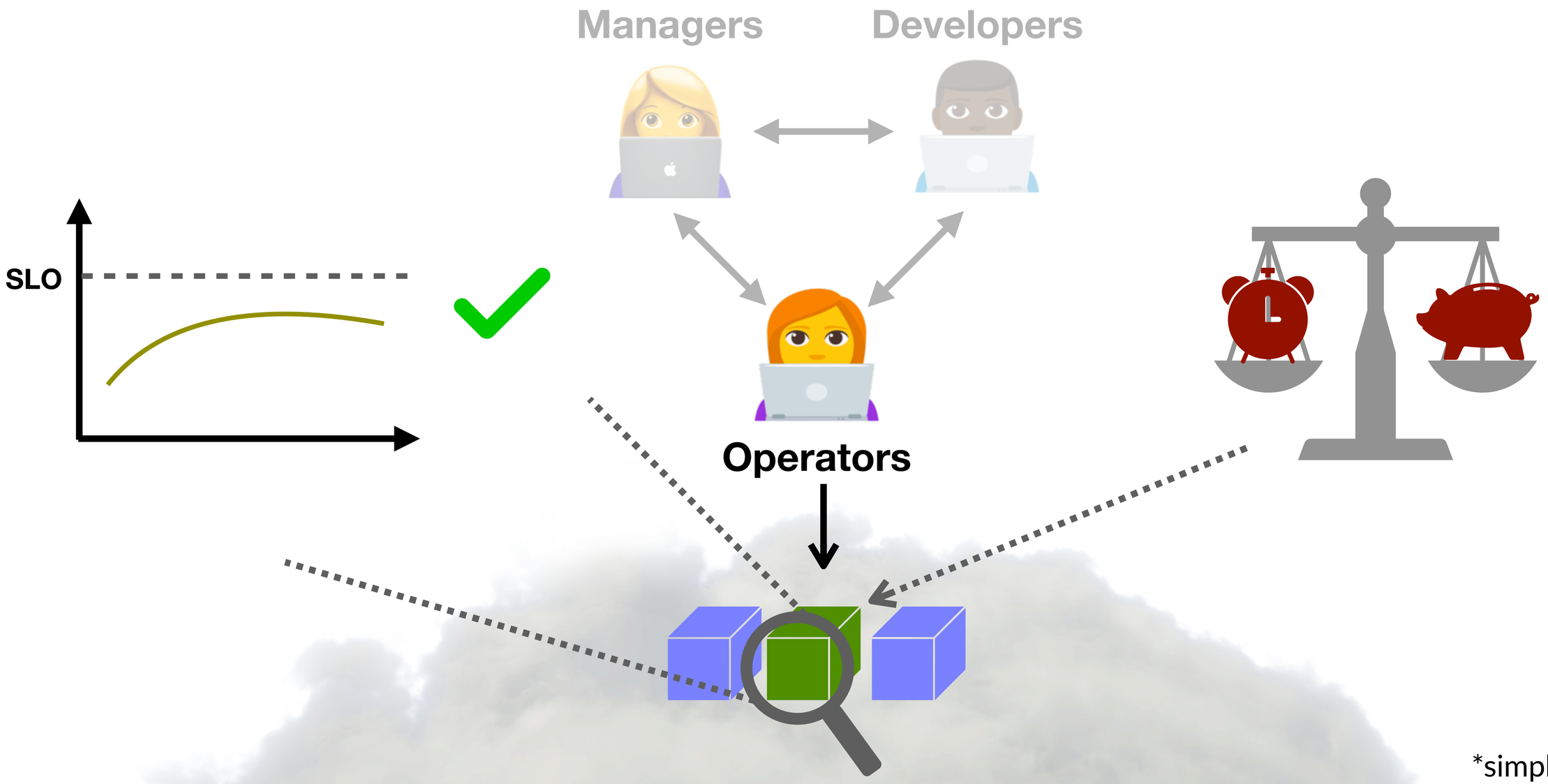

*simplified model

# Operators monitor and make trade-offs



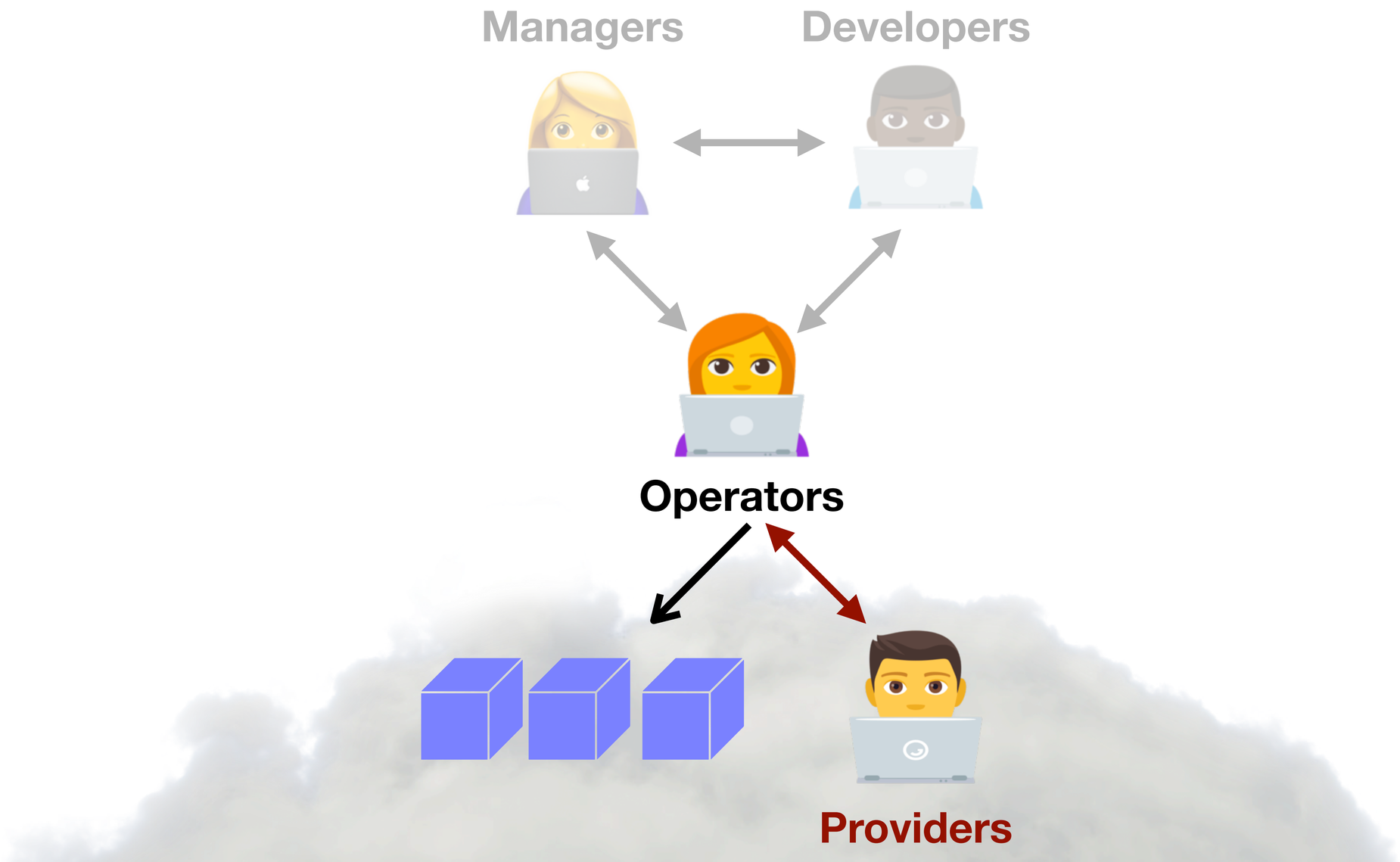

*simplified model

# Operators monitor and make trade-offs to meet service objectives



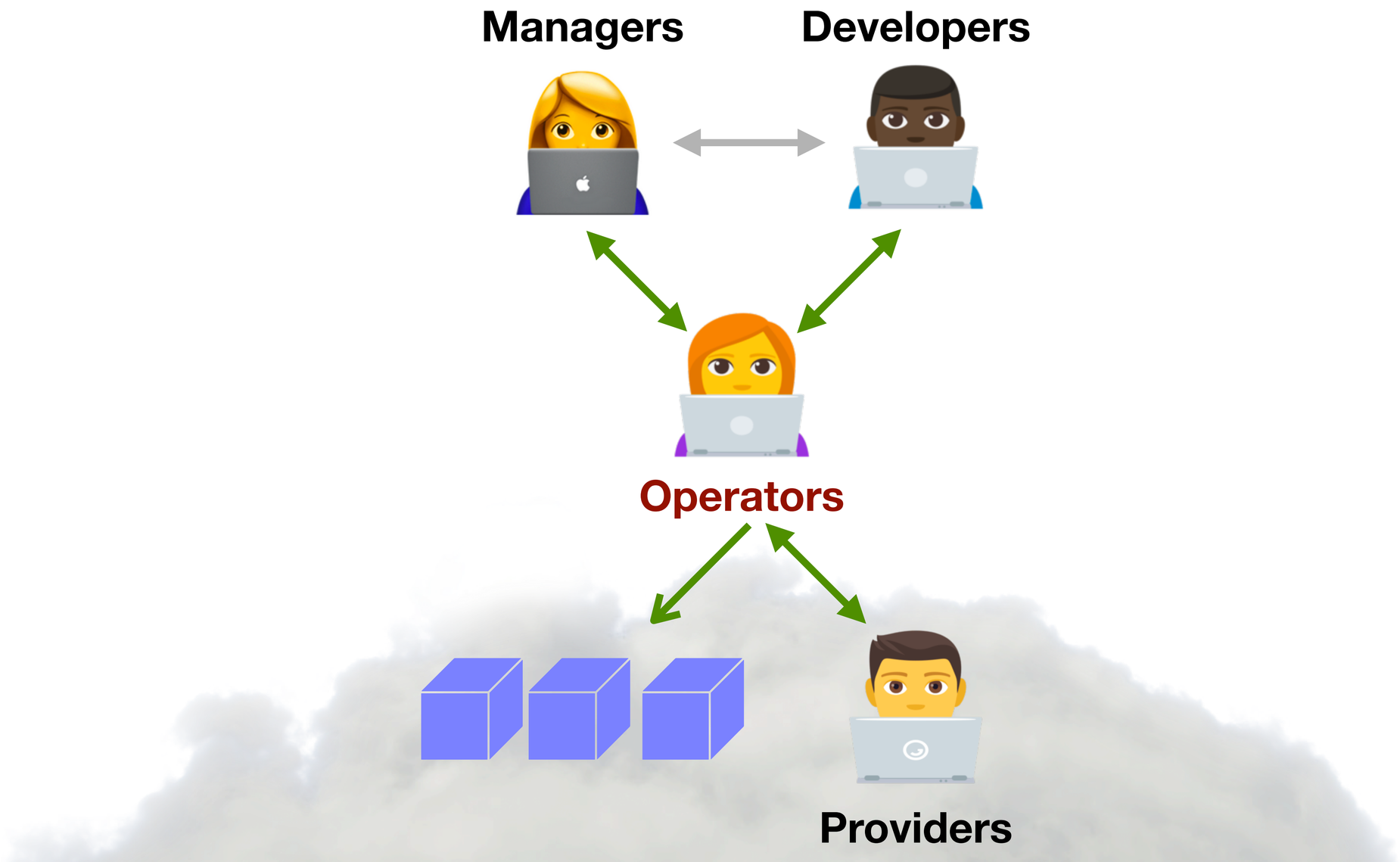

*simplified model

# Operators also work closely with cloud provider teams




*simplified model

# Operators are the narrow waist to enable cloud carbon awareness!

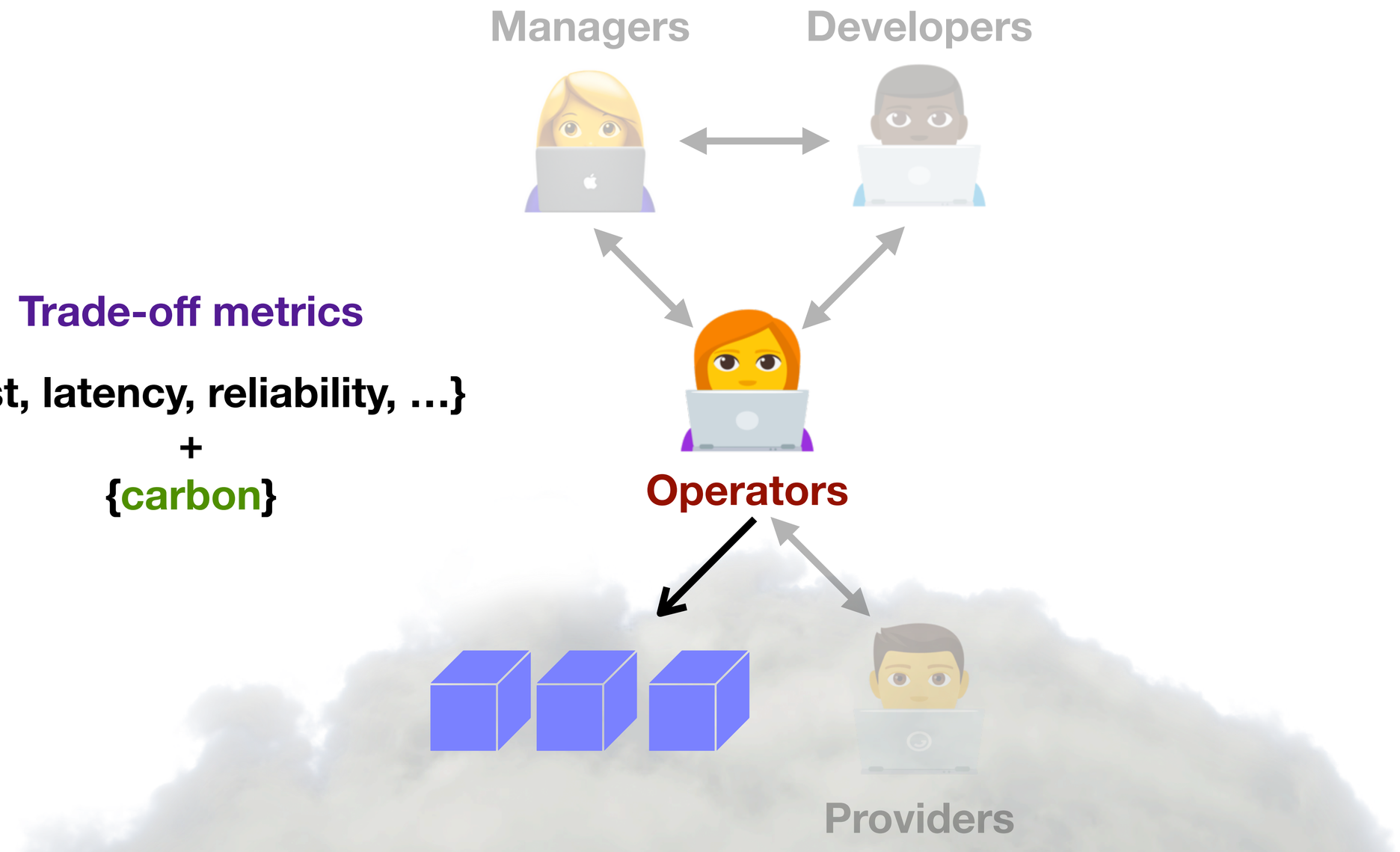



*simplified model

# Viewing carbon as an operations concern

**Trade-off metrics**

**{cost, latency, reliability, …}**
**+**
**{carbon}**

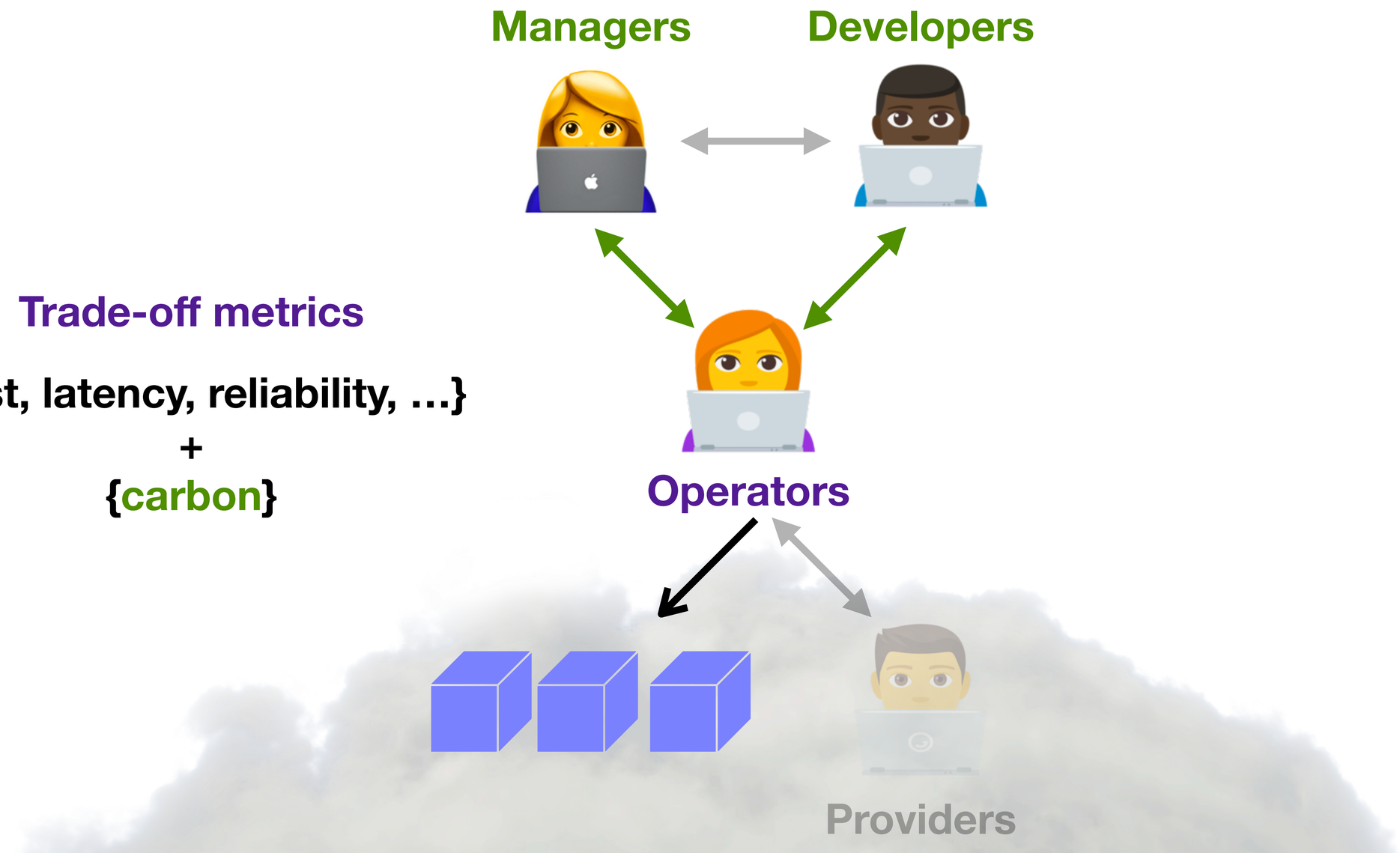



*simplified model

# Operators can help managers and developers be carbon aware

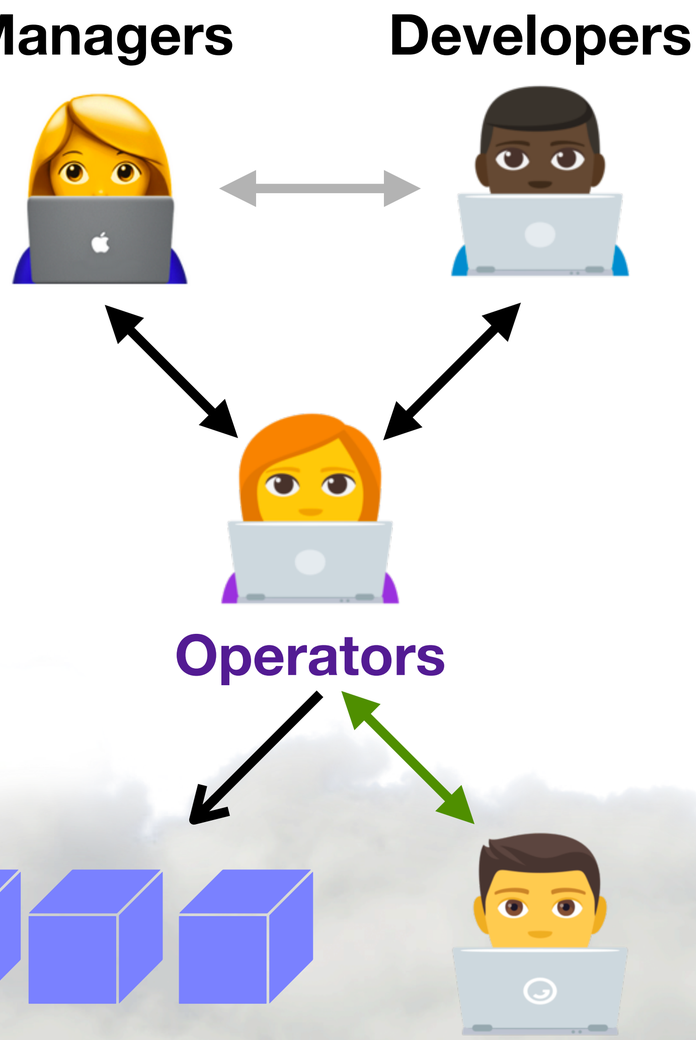
**Trade-off metrics**

**{cost, latency, reliability, …}**
**+**
**{carbon}**




*simplified model

# Operators and providers can communicate visibility to reduce carbon




**Trade-off metrics**

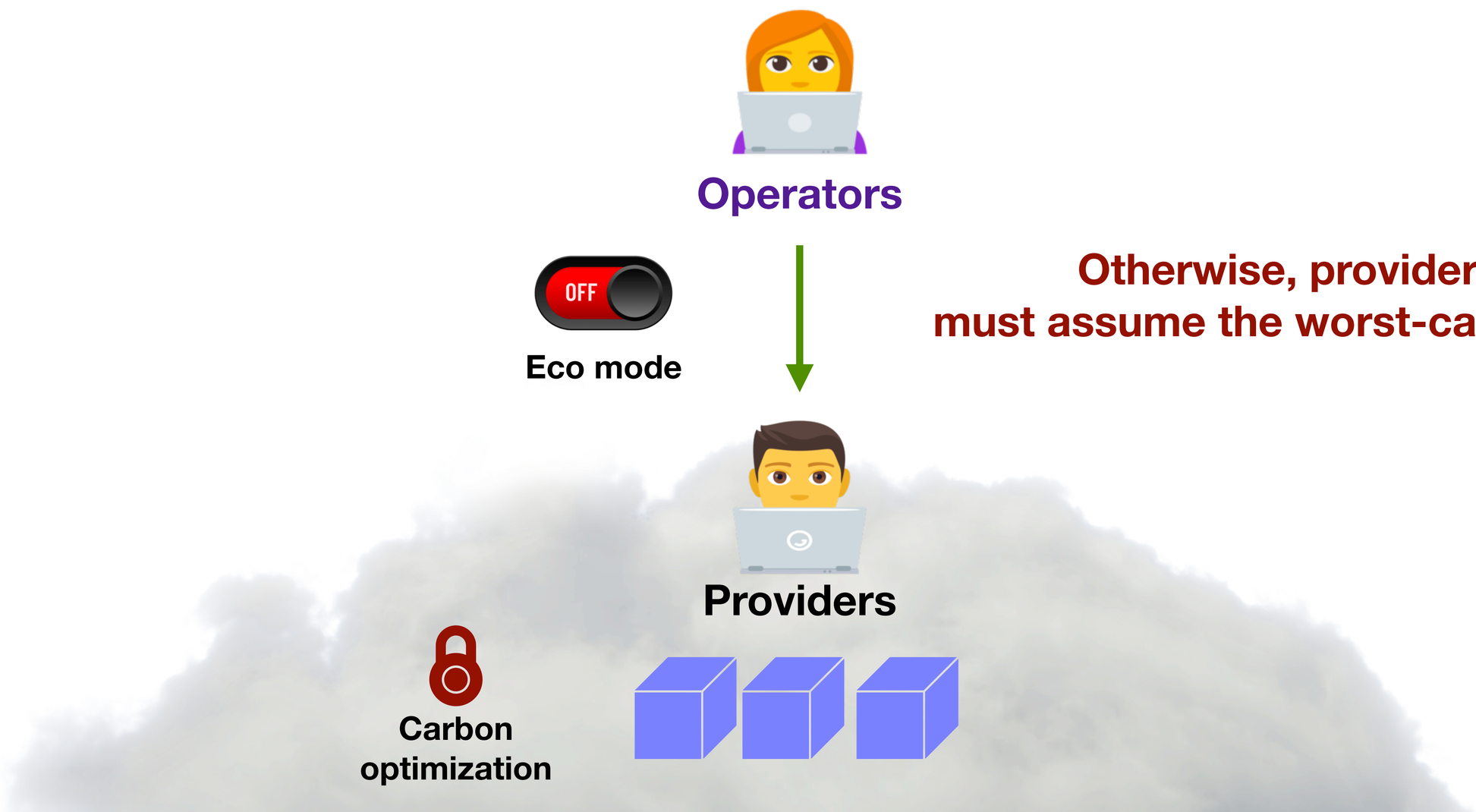
**{cost, latency, reliability, …}**
**+**
**{carbon}**

**Visibility is needed in both directions!**

*simplified model

# Operator—>provider communication with application “eco modes”

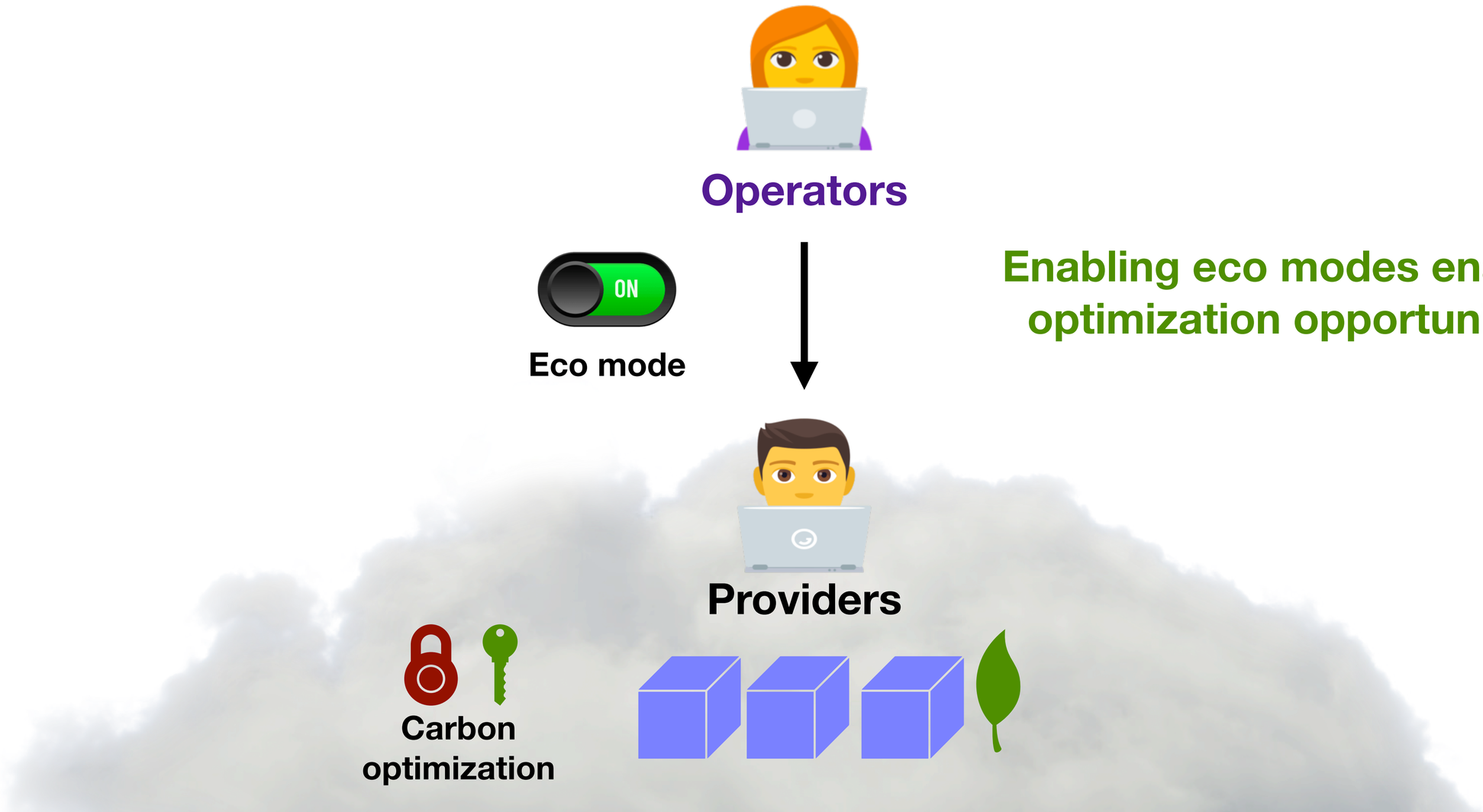
***capture coarse-grained notions of carbon trade-offs***




**Otherwise, providers must assume the worst-case SLOs**

# Toggle applications between power mode and eco mode

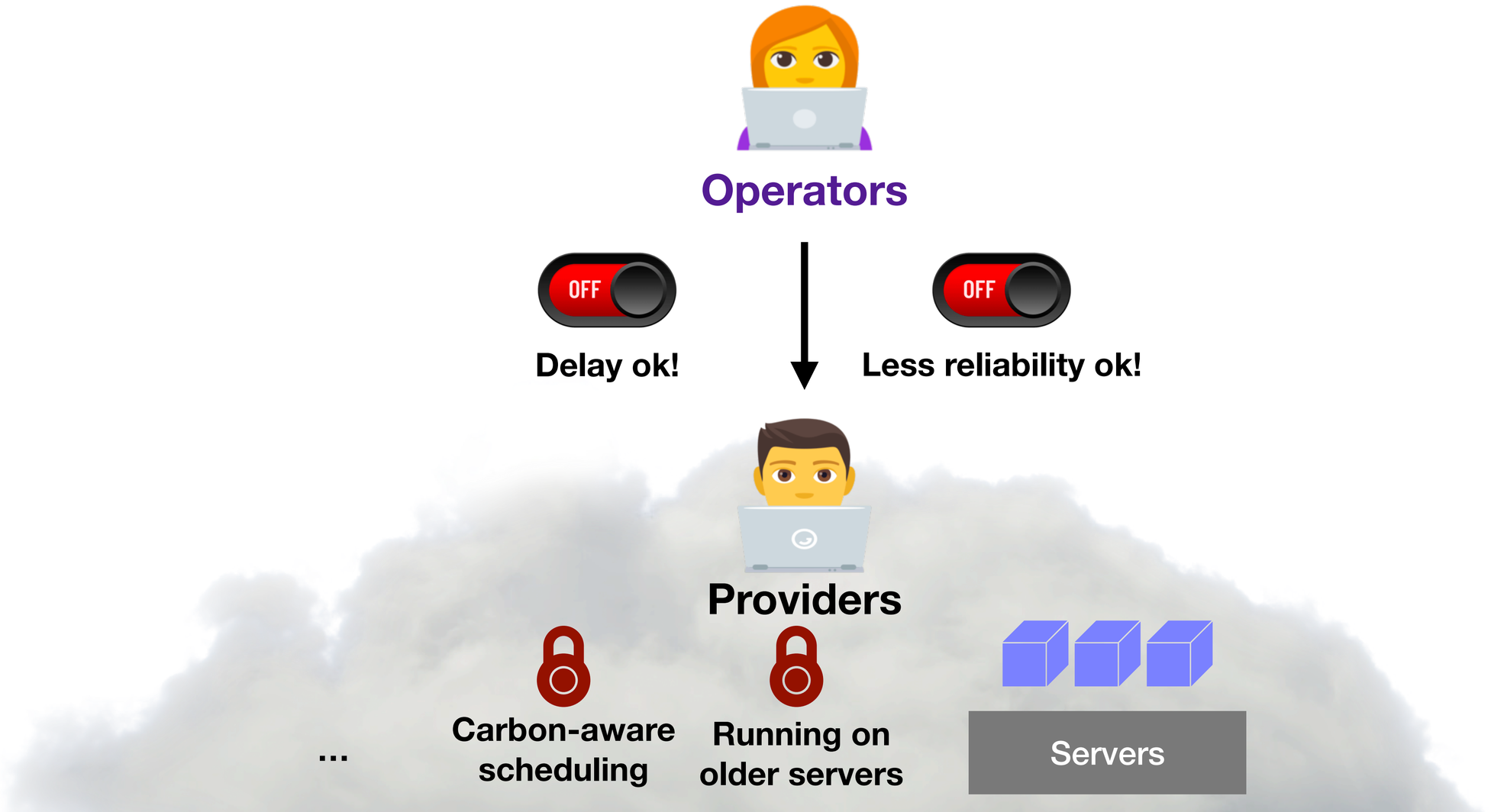
***capture coarse-grained notions of carbon trade-offs***




**Enabling eco modes enables optimization opportunities**

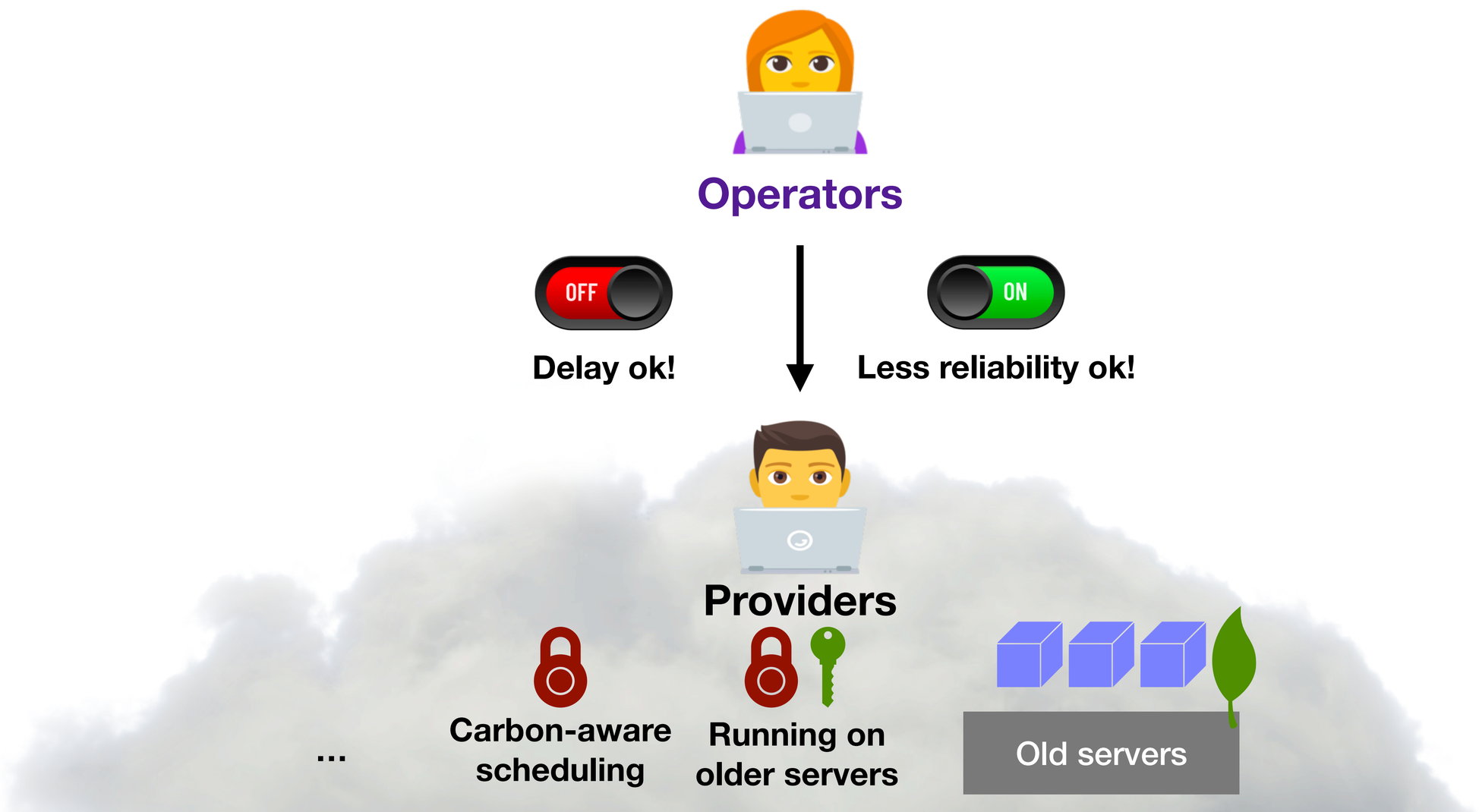
# Different eco modes, different trade-offs

***capture coarse-grained notions of carbon trade-offs***



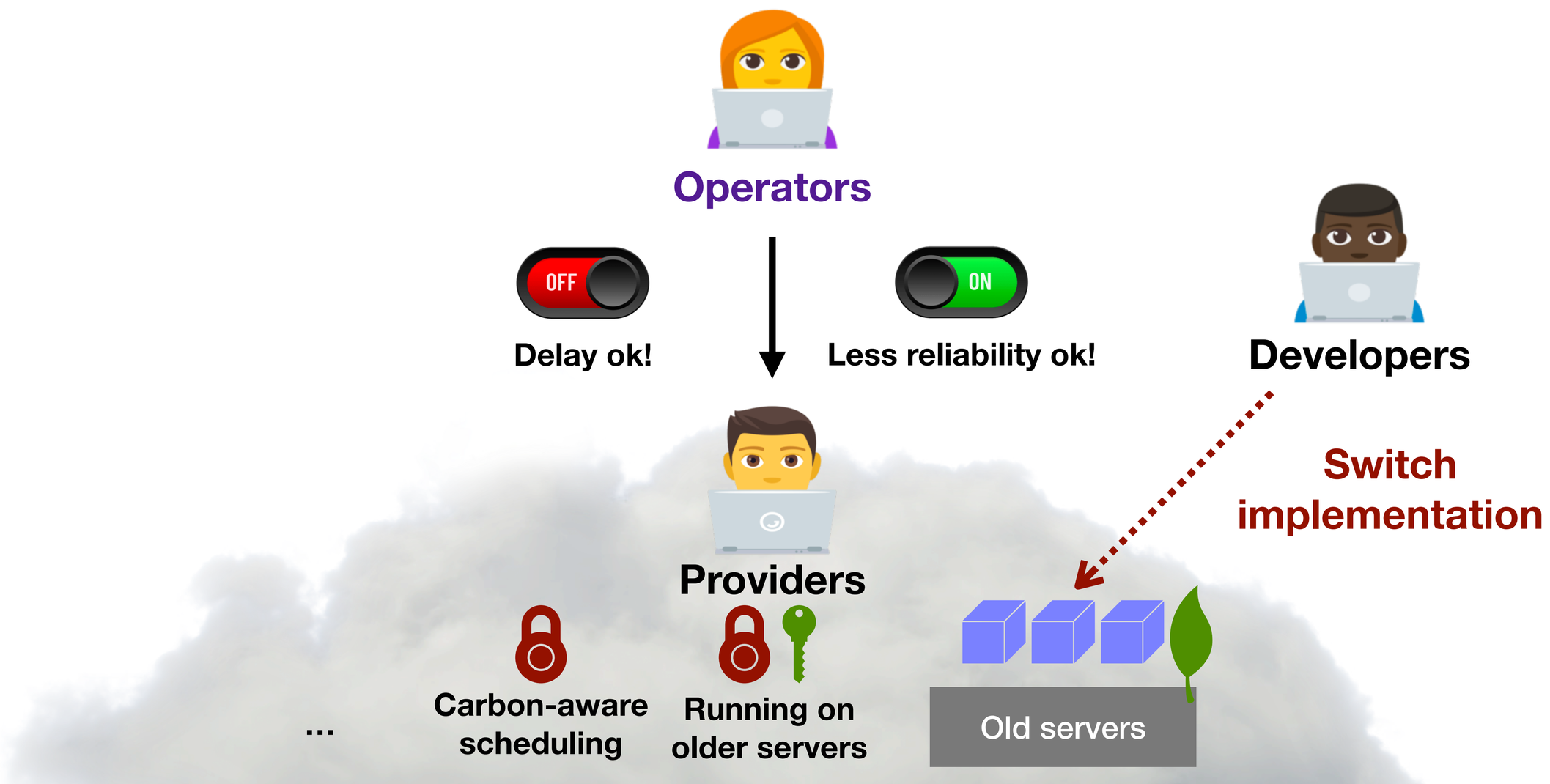
# Different eco modes, different trade-offs

***capture coarse-grained notions of carbon trade-offs***



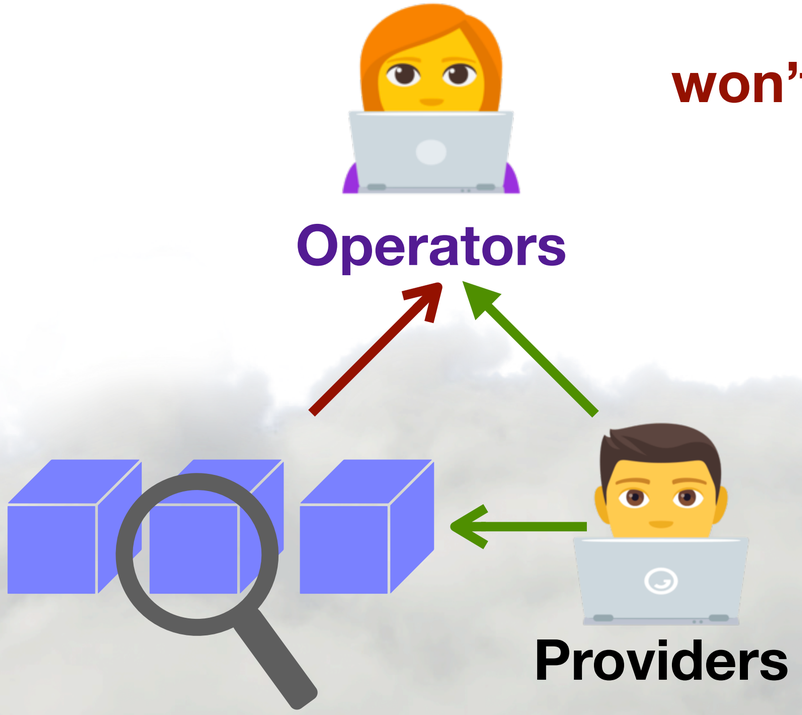
# Developers can extend eco modes by implementing new trade-offs

***capture coarse-grained notions of carbon trade-offs***



# Provider—>operator communication with service-level carbon metrics

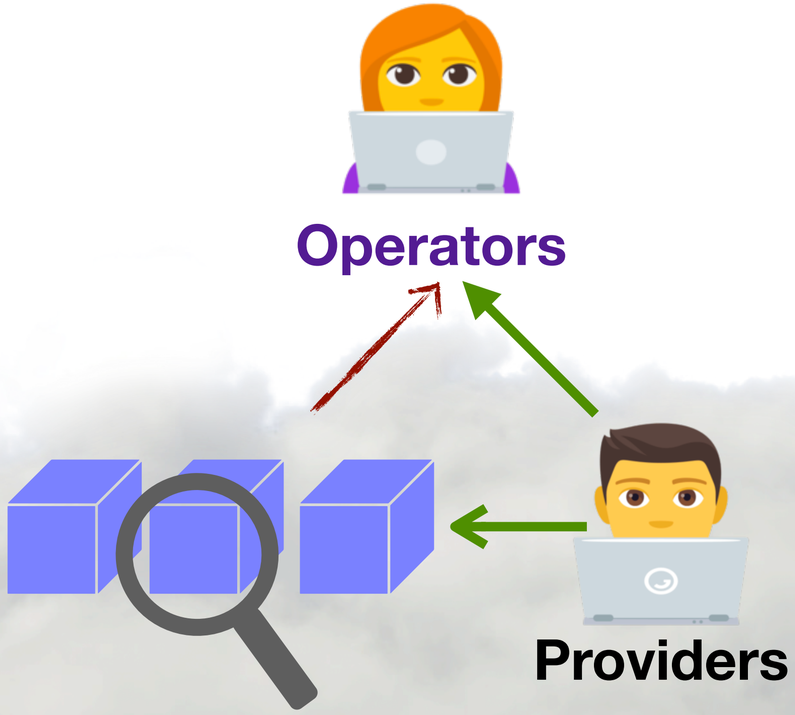
***empower application operators with carbon visibility***




**Otherwise, operators**
**won’t know how to trade-off carbon**

# Approximate metrics are good enough! (for now)

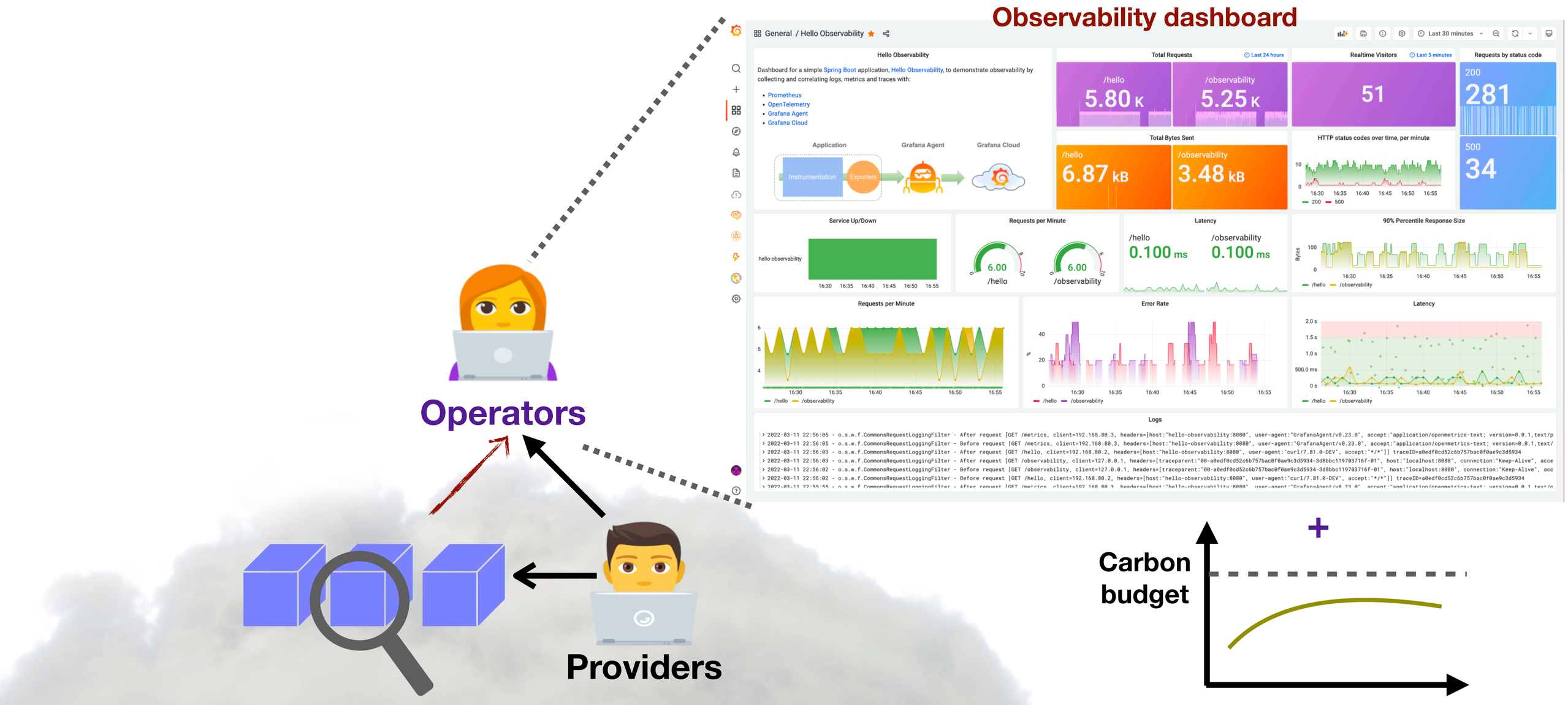
***empower application operators with carbon visibility***




**As long as they enable carbon reduction**

# Integrate carbon into existing operator workflows

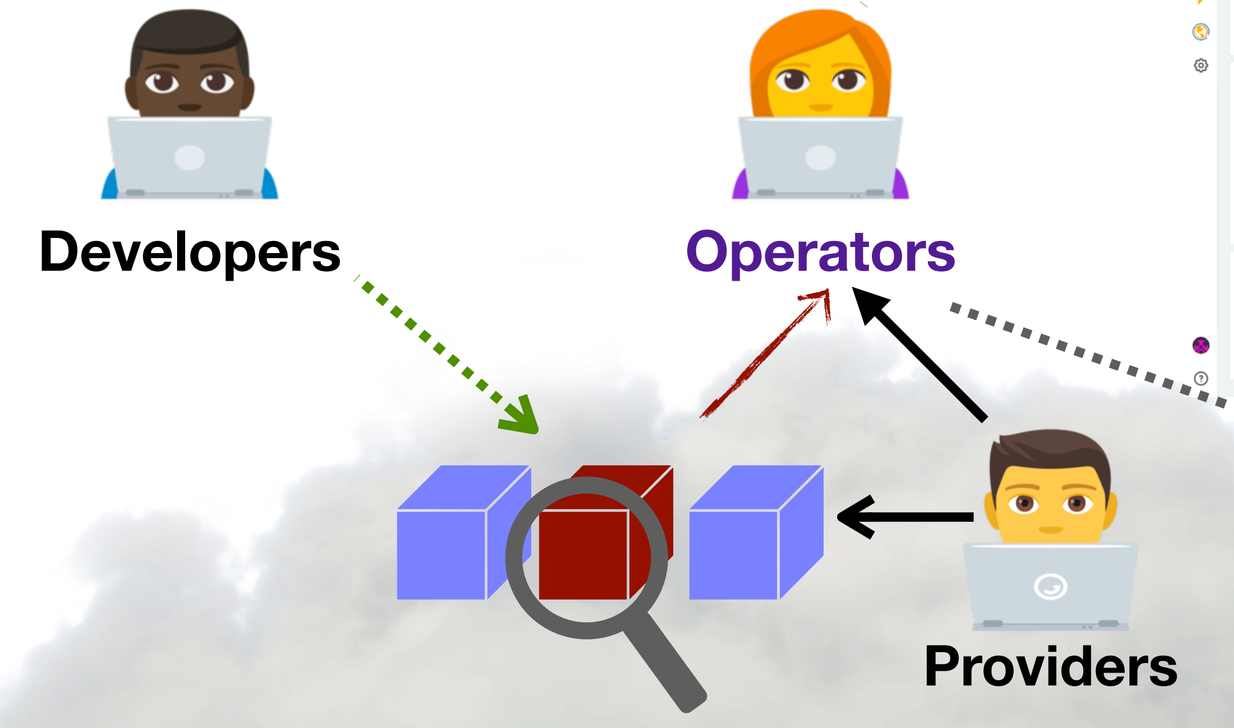
***empower application operators with carbon visibility***

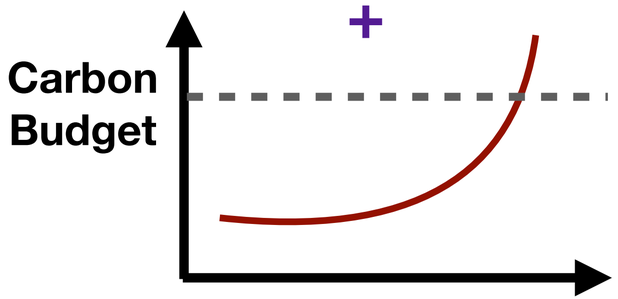


# Helps identify carbon hotspots for developers to optimize

***empower application operators with carbon visibility***



# Carbon knowledge repositories to federate carbon awareness

***"menu cards" for software carbon intensity***

## Key-value stores

| KV-store | Reads | Writes |
|---|---|---|
| **Redis** | 10k req/s | 5k req/s |
| **Memcached** | 12k/s req/s | 4k req/s |

## Deep learning models

| Model | Accuracy | Latency |
|---|---|---|
| **CatGPT** | 99% | 64 ms/token |
| **BatGPT** | 90% | 48 ms/token |

*illustrative

# Carbon knowledge repositories to federate carbon awareness

***"menu cards" for software carbon intensity***

## Key-value stores

| KV-store | Reads | Writes | Carbon |
|---|---|---|---|
| **Redis** | 10k req/s | 5k req/s | **??** |
| **Memcached** | 12k/s req/s | 4k req/s | **??** |

## Deep learning models

| Model | Accuracy | Latency | Carbon |
|---|---|---|---|
| **CatGPT** | 99% | 64 ms/token | **??** |
| **BatGPT** | 90% | 48 ms/token | **??** |

**Standardize carbon benchmarks**

*illustrative

# Carbon knowledge repositories to federate carbon awareness

***"menu cards" for software carbon intensity***

## Container registry

| Image | Reads | Writes | Carbon |
|---|---|---|---|
| **Redis** | 10k req/s | 5k req/s | **??** |
| **Memcached** | 12k/s req/s | 4k req/s | **??** |

## Model zoo

| Model | Accuracy | Latency | Carbon |
|---|---|---|---|
| **CatGPT** | 99% | 64 ms/token | **??** |
| **BatGPT** | 90% | 48 ms/token | **??** |

**Integrate into popular repositories**

**Standardize carbon benchmarks**

*illustrative

# Carbon knowledge repositories to federate carbon awareness

***"menu cards" for software carbon intensity***

## Container registry

| Image | Reads | Writes | Carbon |
|---|---|---|---|
| **Redis** | 10k req/s | 5k req/s | **??** |
| **Memcached** | 12k/s req/s | 4k req/s | **??** |

## Model zoo

| Model | Accuracy | Latency | Carbon |
|---|---|---|---|
| **CatGPT** | 99% | 64 ms/token | **??** |
| **BatGPT** | 90% | 48 ms/token | **??** |

**Check out: https://ml.energy/leaderboard/**

**Integrate into popular repositories**

**Standardize carbon benchmarks**

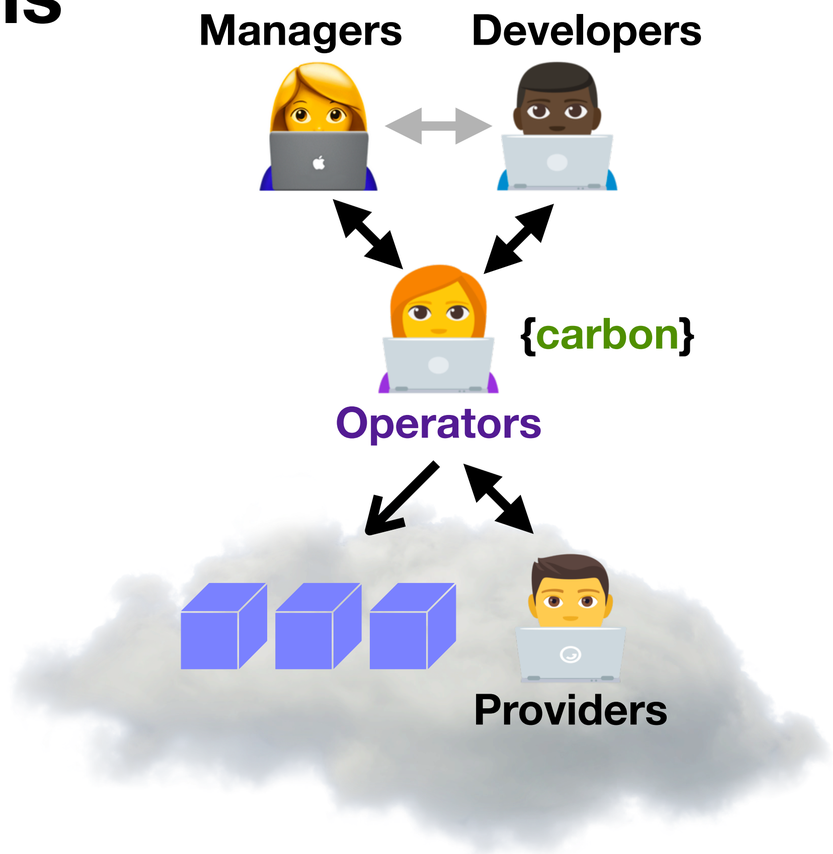
*illustrative

# An agile pathway towards carbon-aware clouds

1. **Federate carbon responsibility and tools** with operators as the narrow waist

2. **Provide actionable visibility** into carbon emissions

3. **Centralize configurable optimizations** for SLO-aware carbon reduction




Thanks!

pratyush@cs.uw.edu