Adding intelligence to stream-processing applications

Michael McCune and William Benton
msm@redhat.com • @FOSSJunkie
willb@redhat.com • @willb

Forecast

Introducing intelligent applications

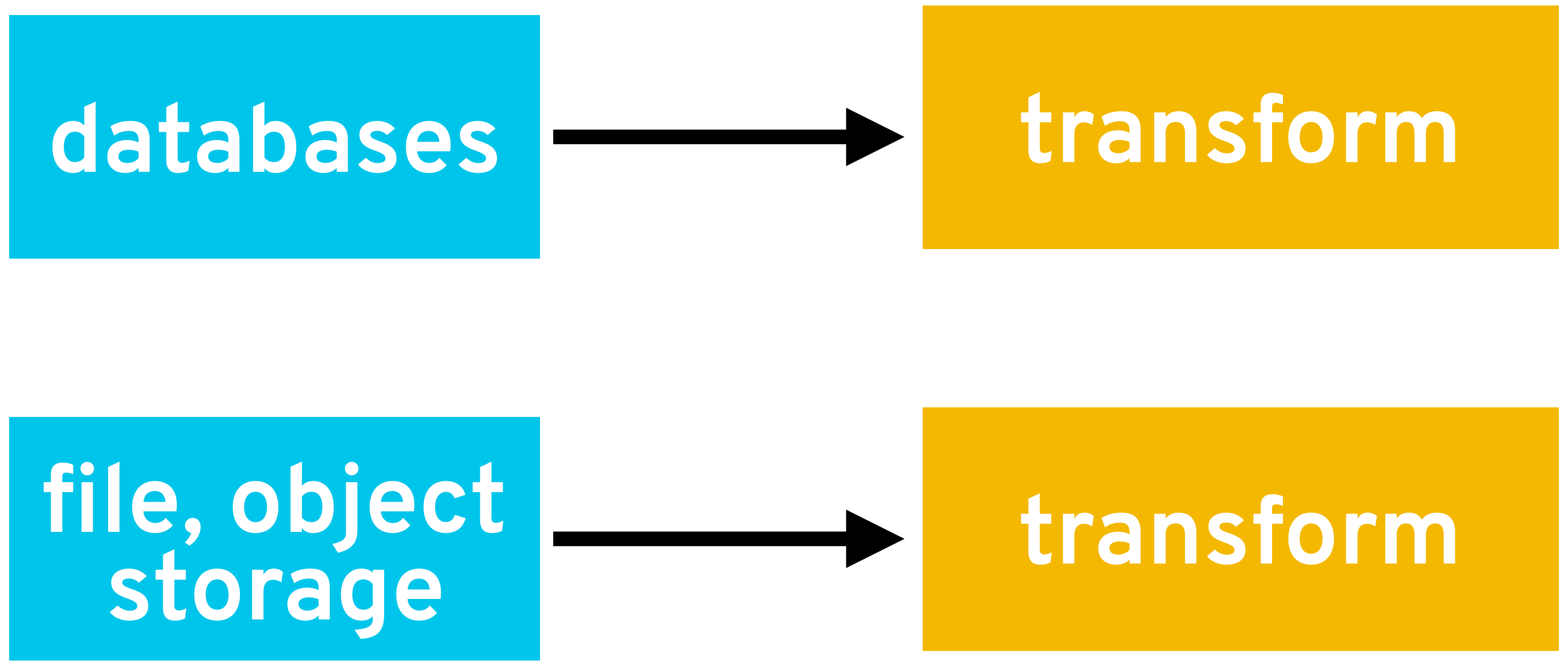
Introducing microservice architectures

Stream processing with Apache Kafka and Apache Spark

Interactivity time

Intelligent applications

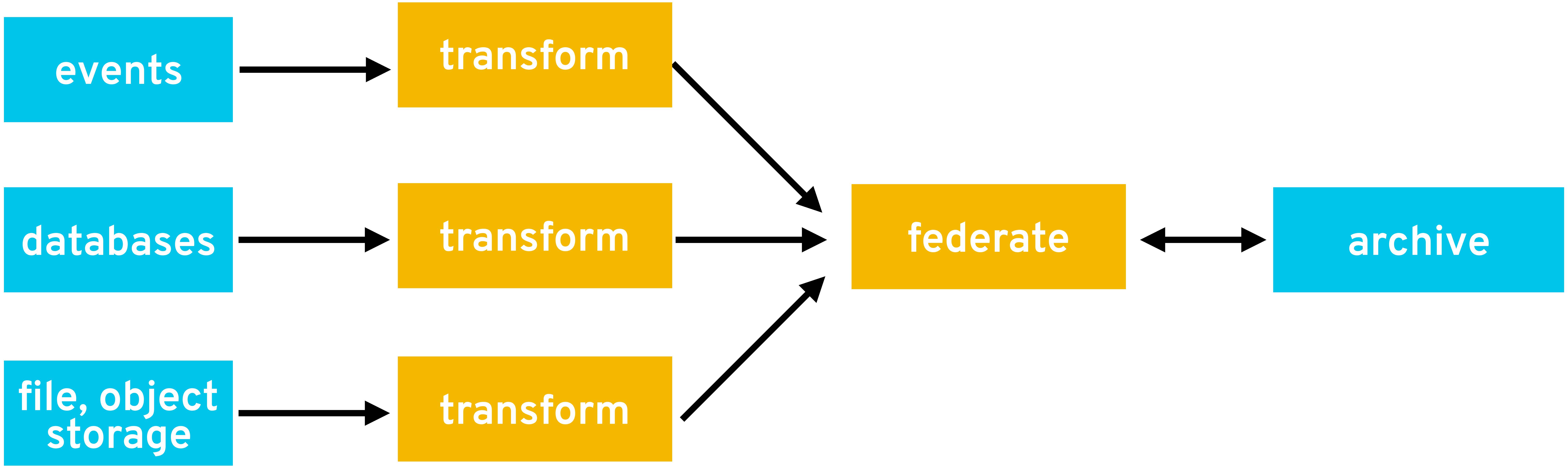
Intelligent applications **collect** and **learn from** data in order to provide **improved functionality** with **longevity** and **popularity**.

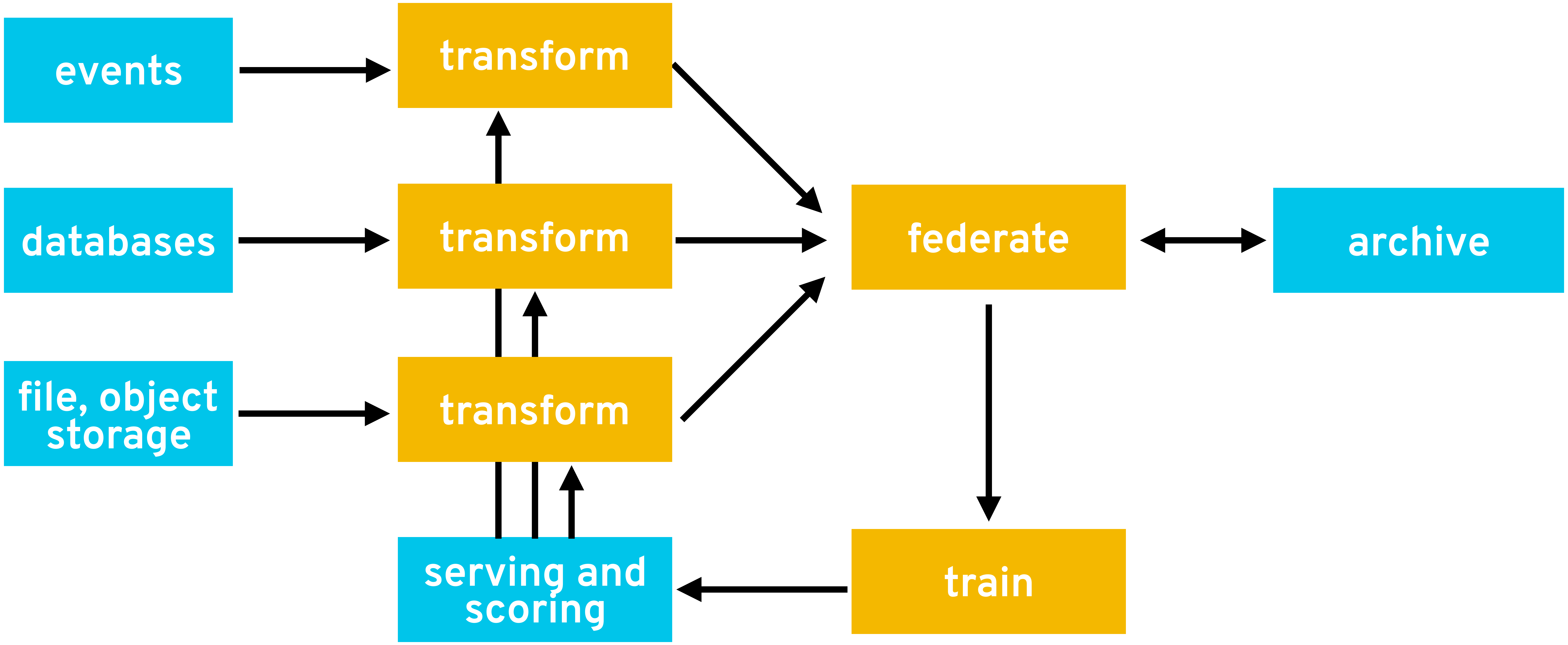

databases

transform

file, object storage

transform


events
databases
file, object storage
transform
transform
transform
federate
archive



events

databases

file, object storage

transform

transform

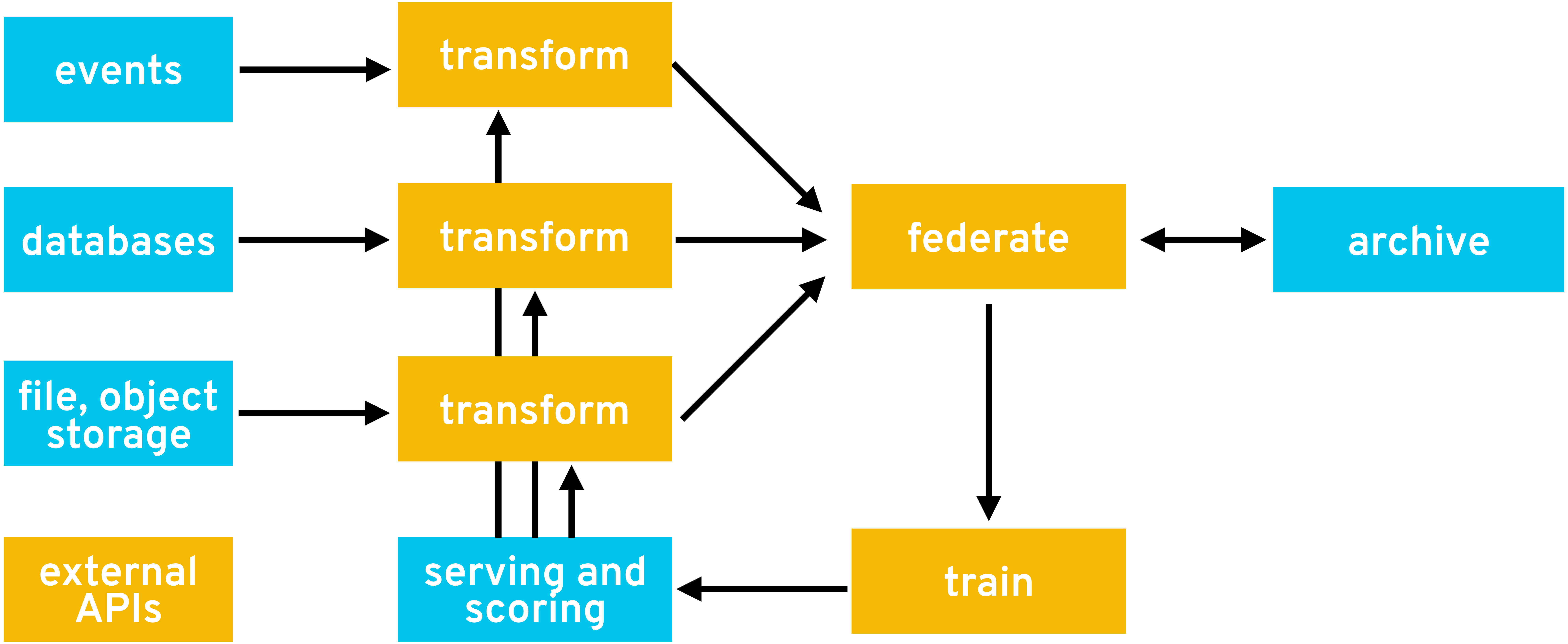
transform

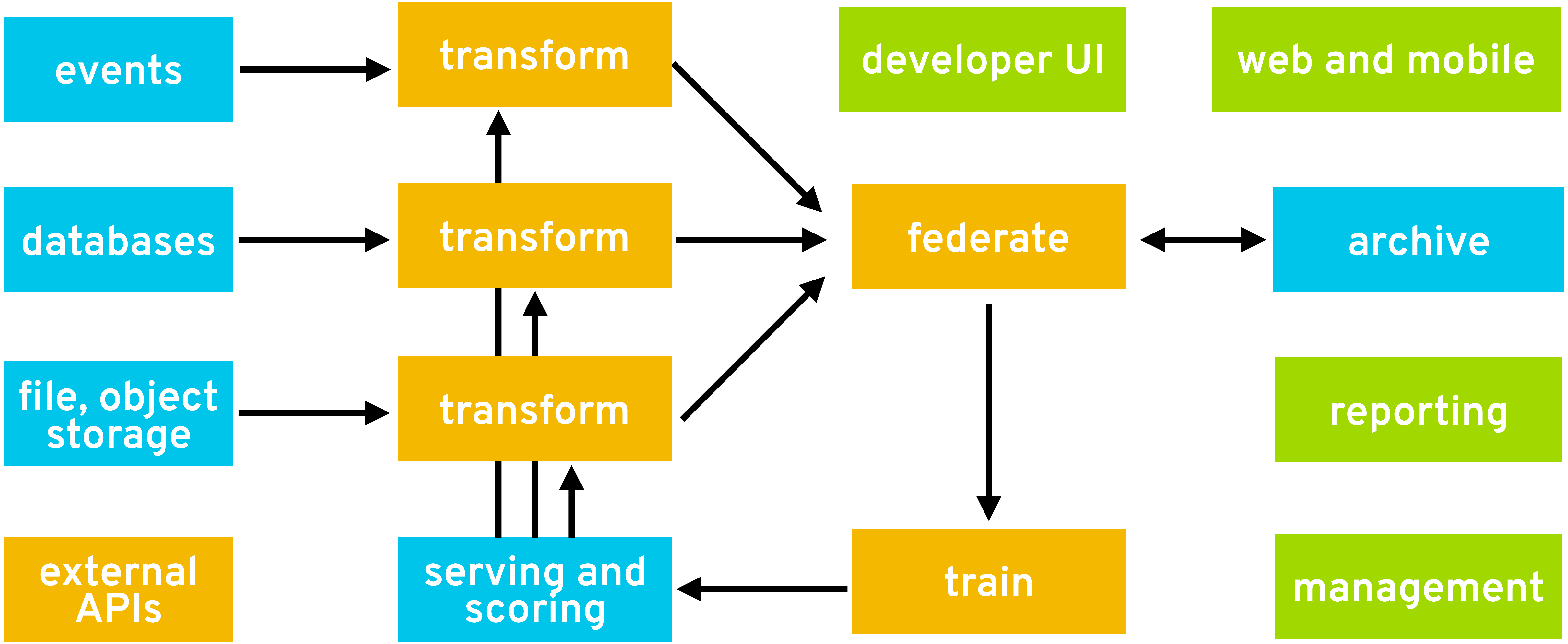
serving and scoring

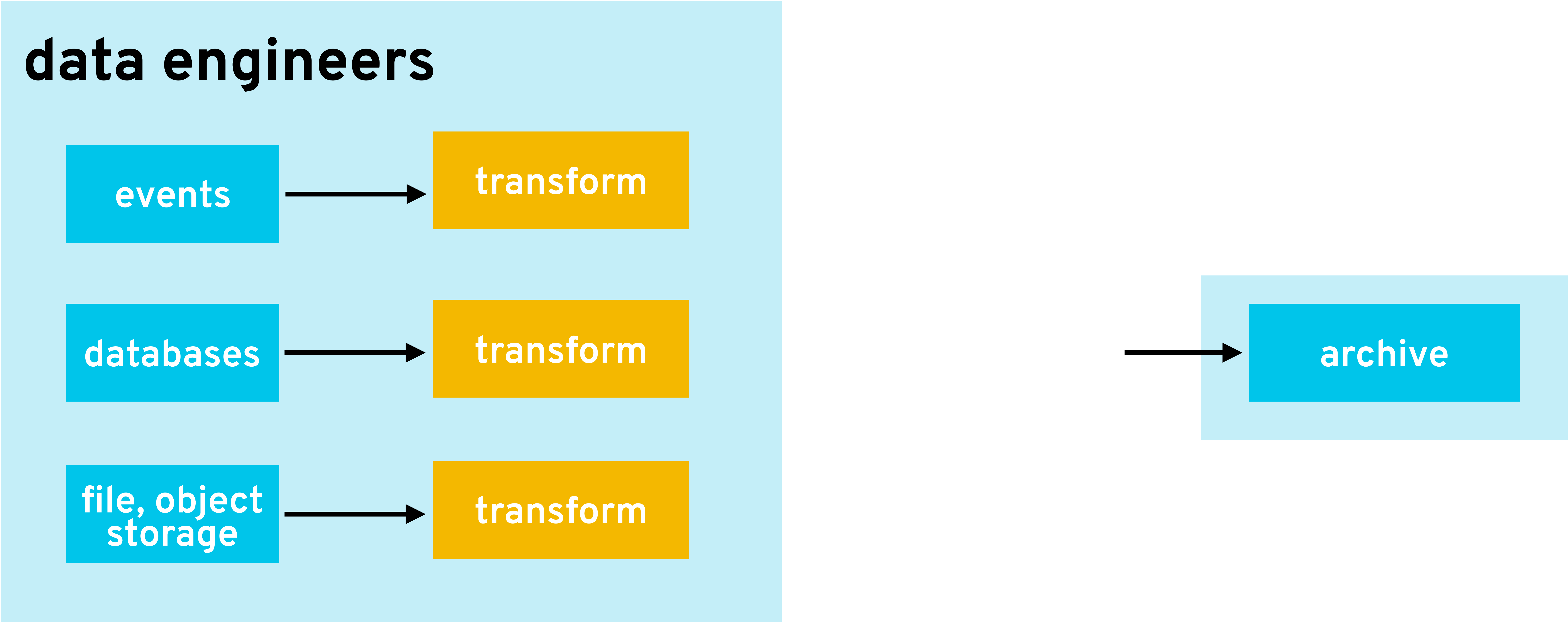
federate

archive

train


events
databases
file, object storage
external APIs
transform
transform
transform
serving and scoring
federate
archive
train


events
transform
developer UI
web and mobile
databases
transform
federate
archive
file, object storage
transform
reporting
external APIs
serving and scoring
train
management



data engineers

events

databases

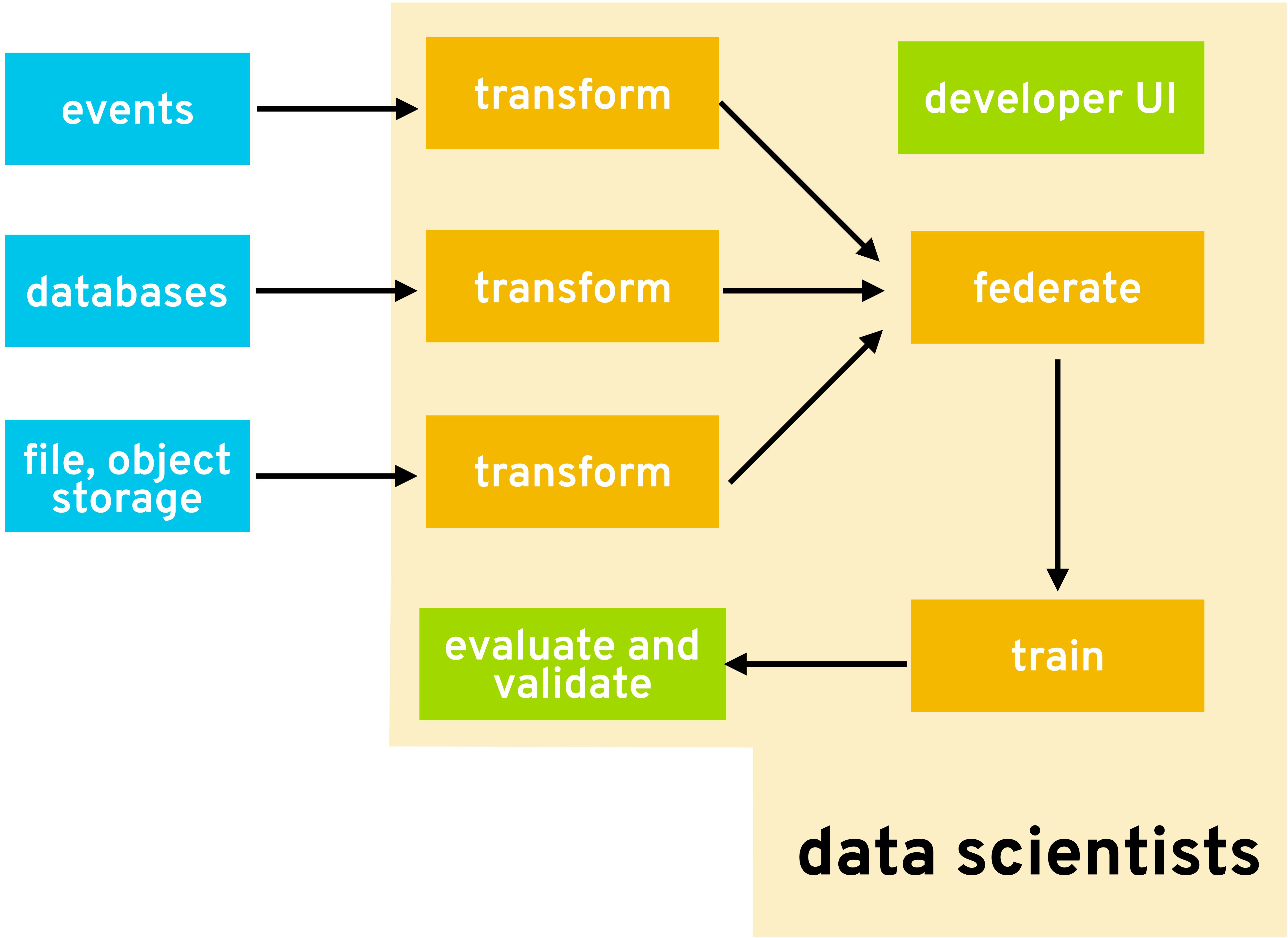
file, object storage

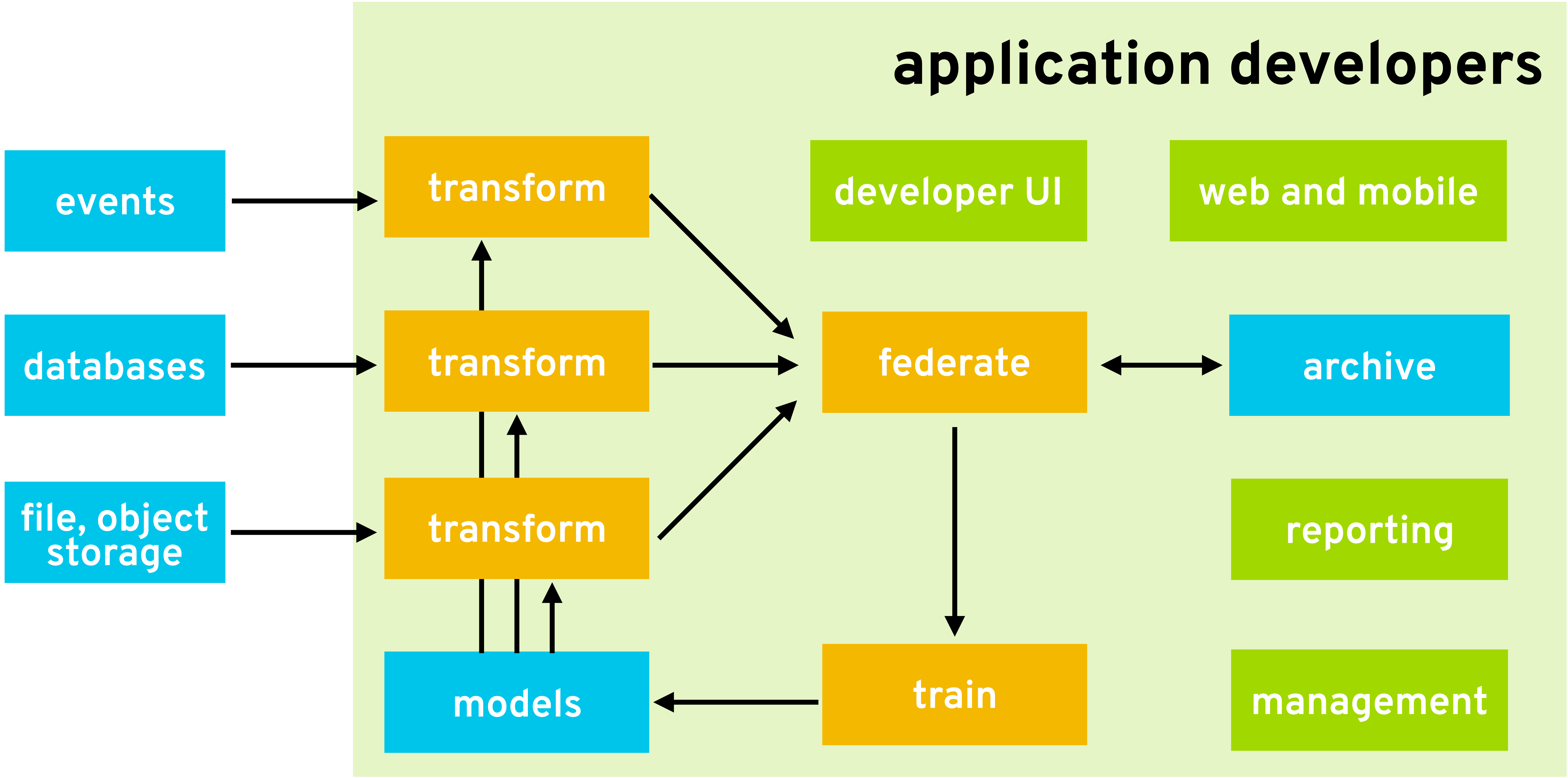
transform

transform

transform

archive


events
databases
file, object storage
transform
transform
transform
developer UI
federate
train
evaluate and validate
data scientists



application developers

events

databases

file, object storage

transform

transform

transform

models

developer UI

web and mobile

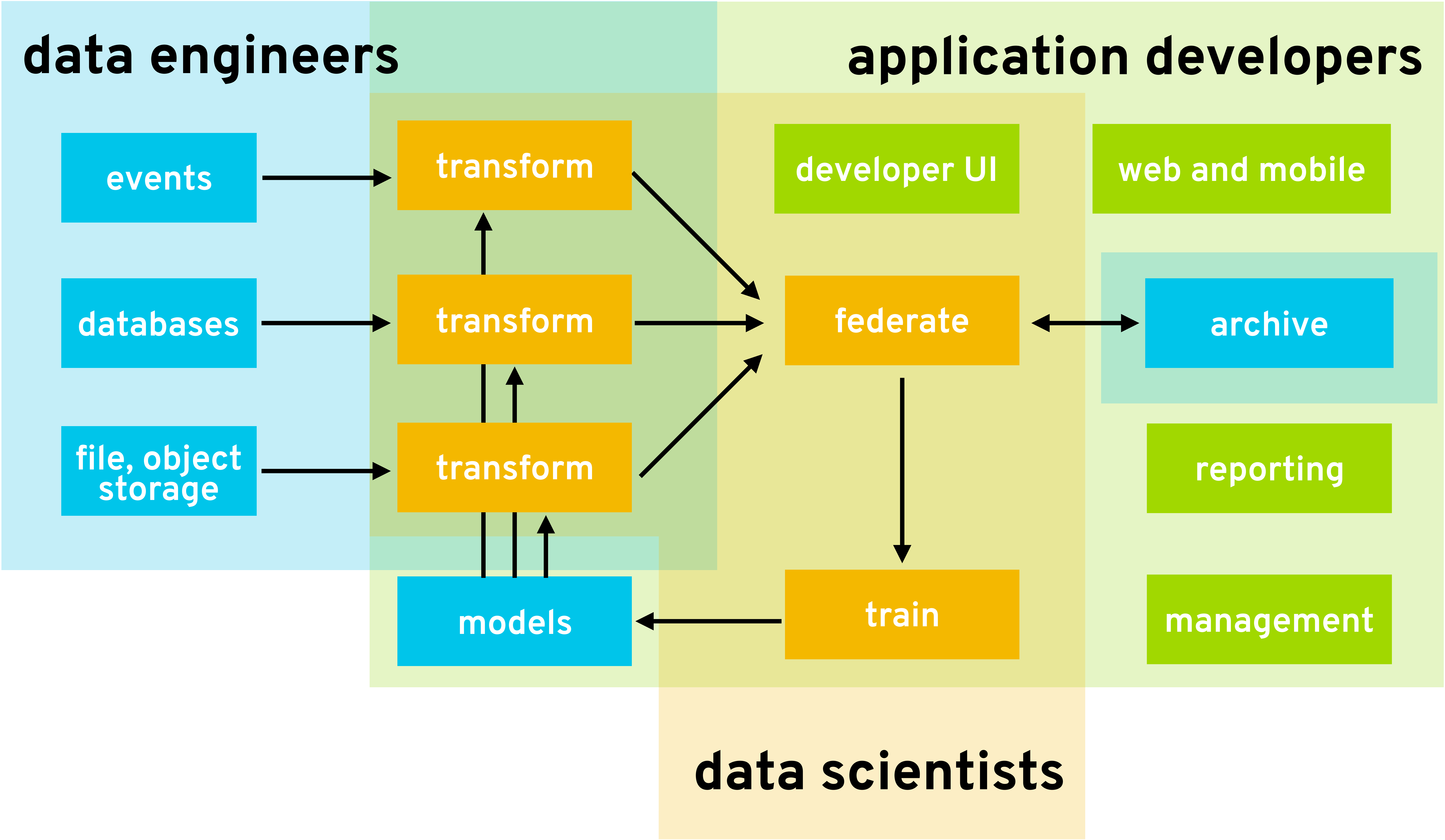
federate

archive

train

reporting

management

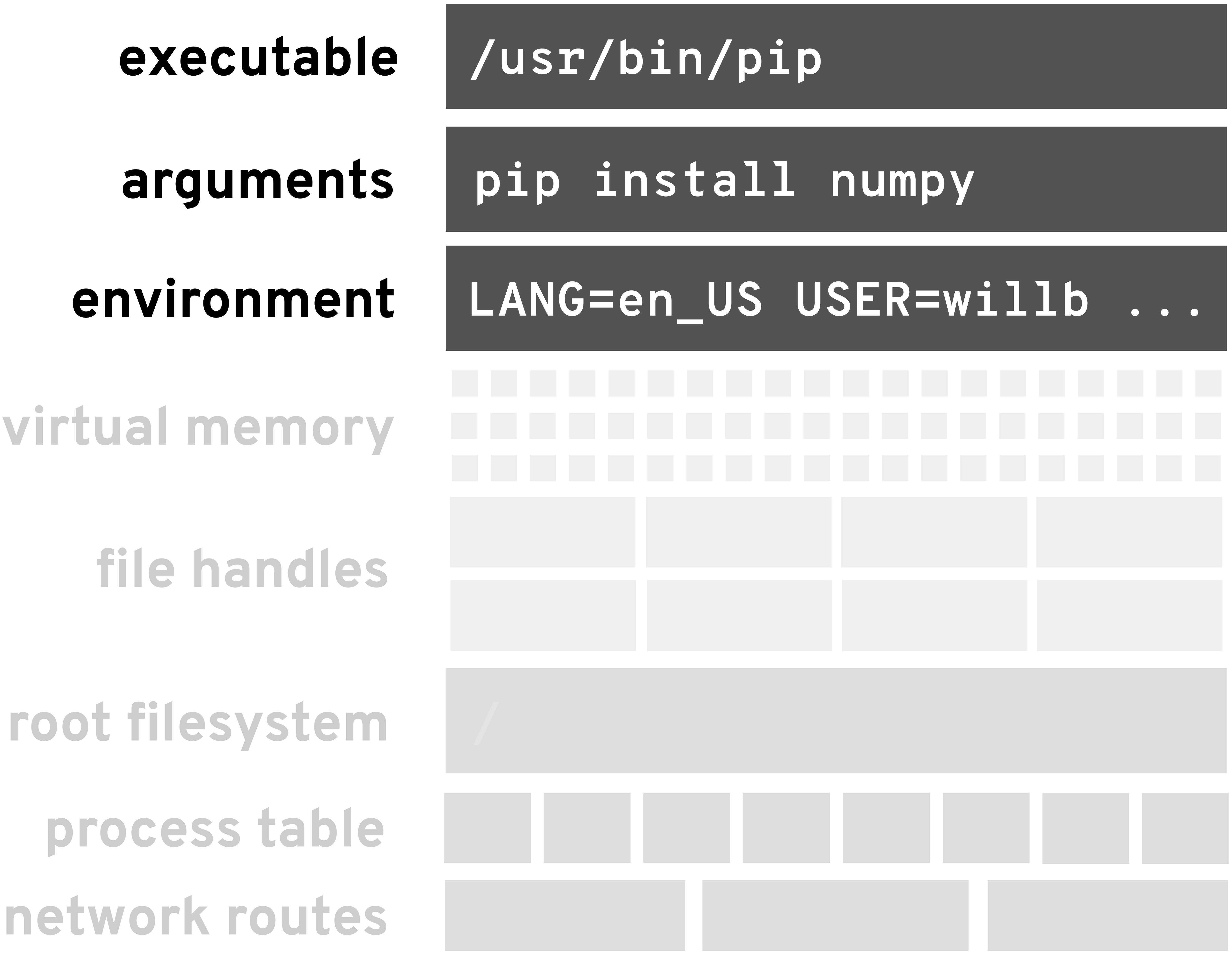

data engineers
application developers
events
transform
developer UI
web and mobile
databases
transform
federate
archive
file, object storage
transform
reporting
models
train
management
data scientists

Containers: great for application developers and data scientists

What is a container?

%

```
% pip install numpy
```



executable `/usr/bin/pip`

arguments `pip install numpy`

environment `LANG=en_US USER=willb ...`

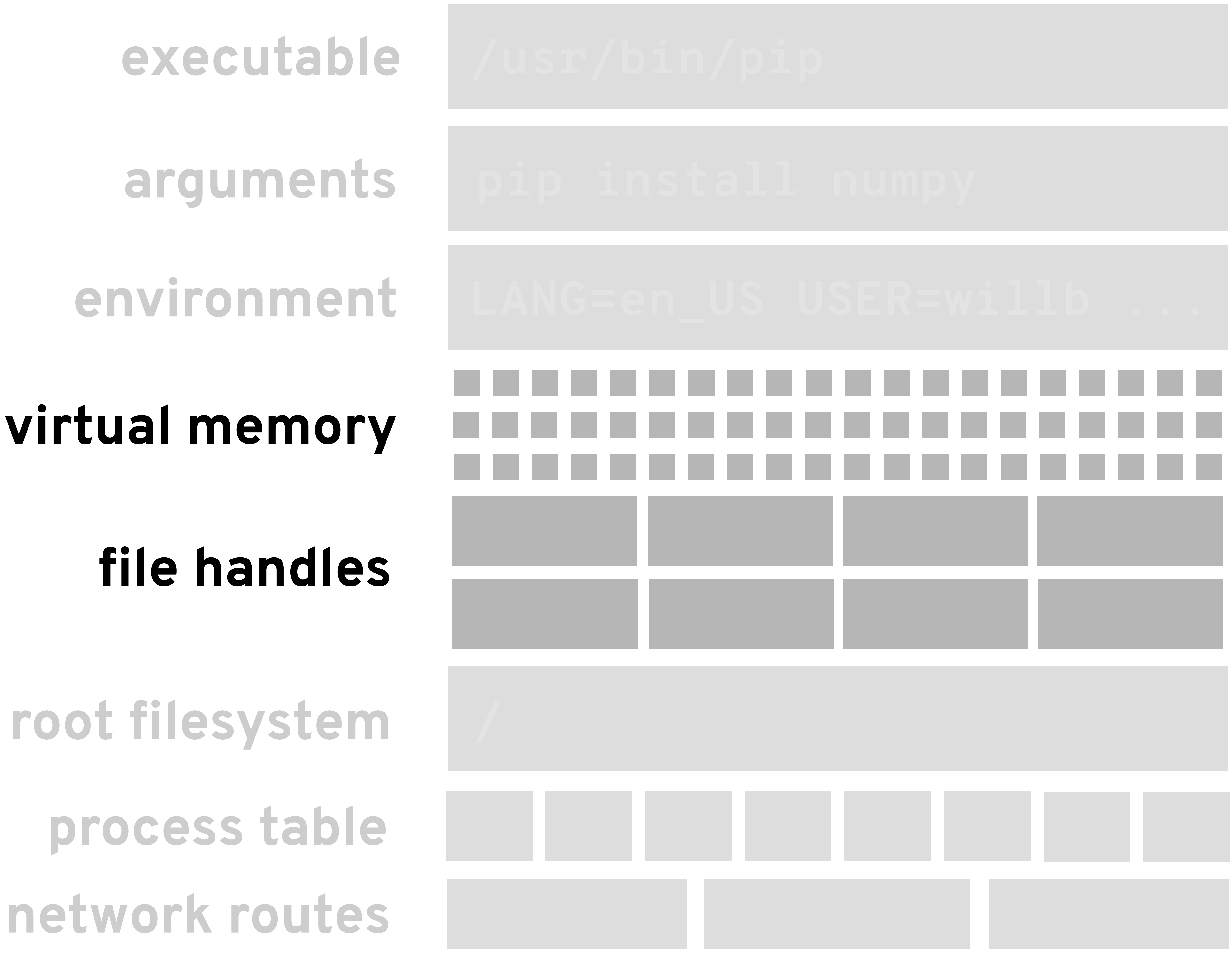
virtual memory

file handles

root filesystem `/`

process table

network routes



executable `/usr/bin/pip`

arguments `pip install numpy`

environment `LANG=en_US USER=willb ...`

virtual memory

file handles

root filesystem `/`

process table

network routes

executable `/usr/bin/pip`

arguments `pip install numpy`

```
Software Failure.   Press left mouse button to continue.
Guru Meditation #00000004.0000AAC0
```

root filesystem `/`

process table

network routes

executable /usr/bin/pip

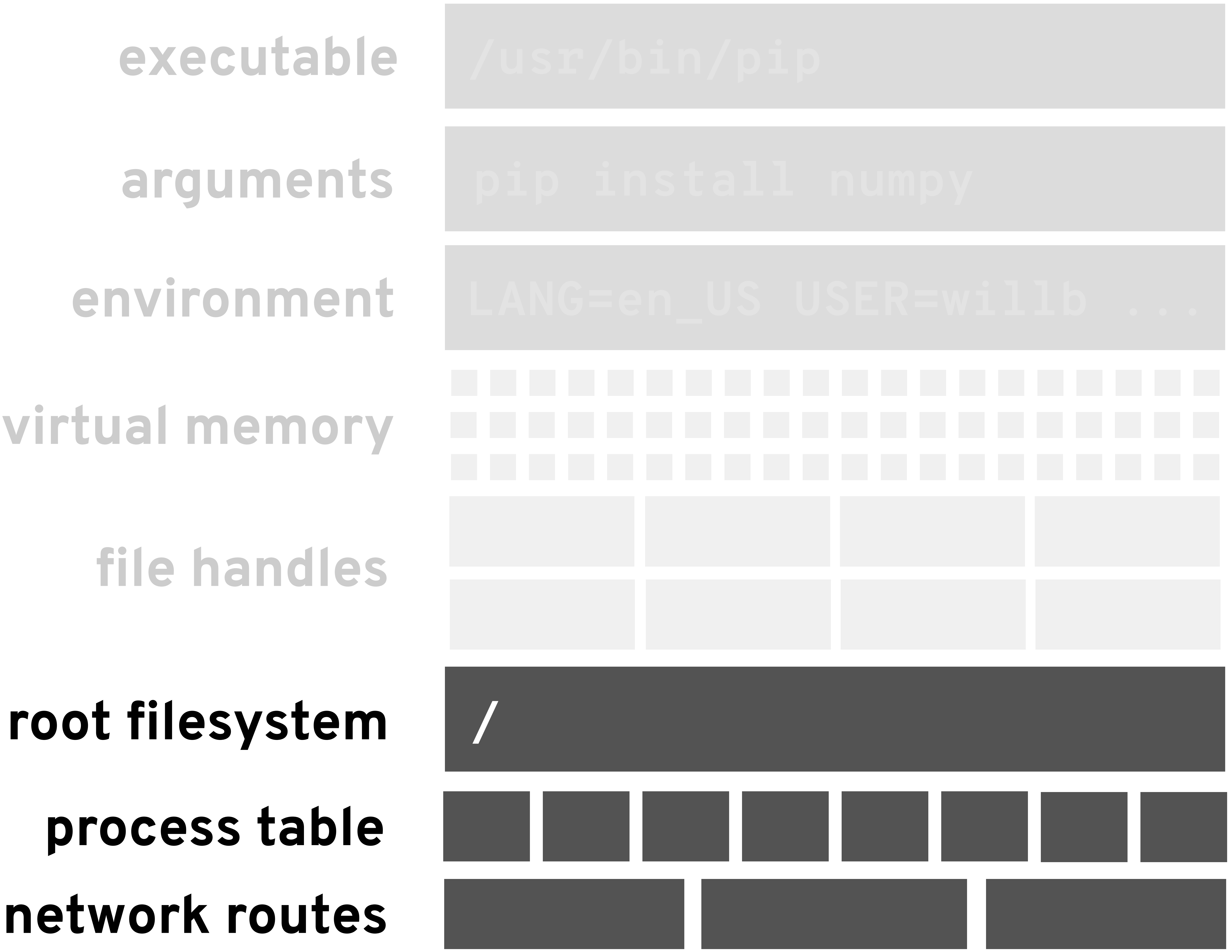
arguments pip install numpy

```
Software Failure.   Press left mouse button to continue.
        Guru Meditation #00000004.0000AAC0
```

root filesystem /

process table

network routes



executable

/usr/bin/pip

arguments

pip install numpy

environment

LANG=en_US USER=willb ...

virtual memory

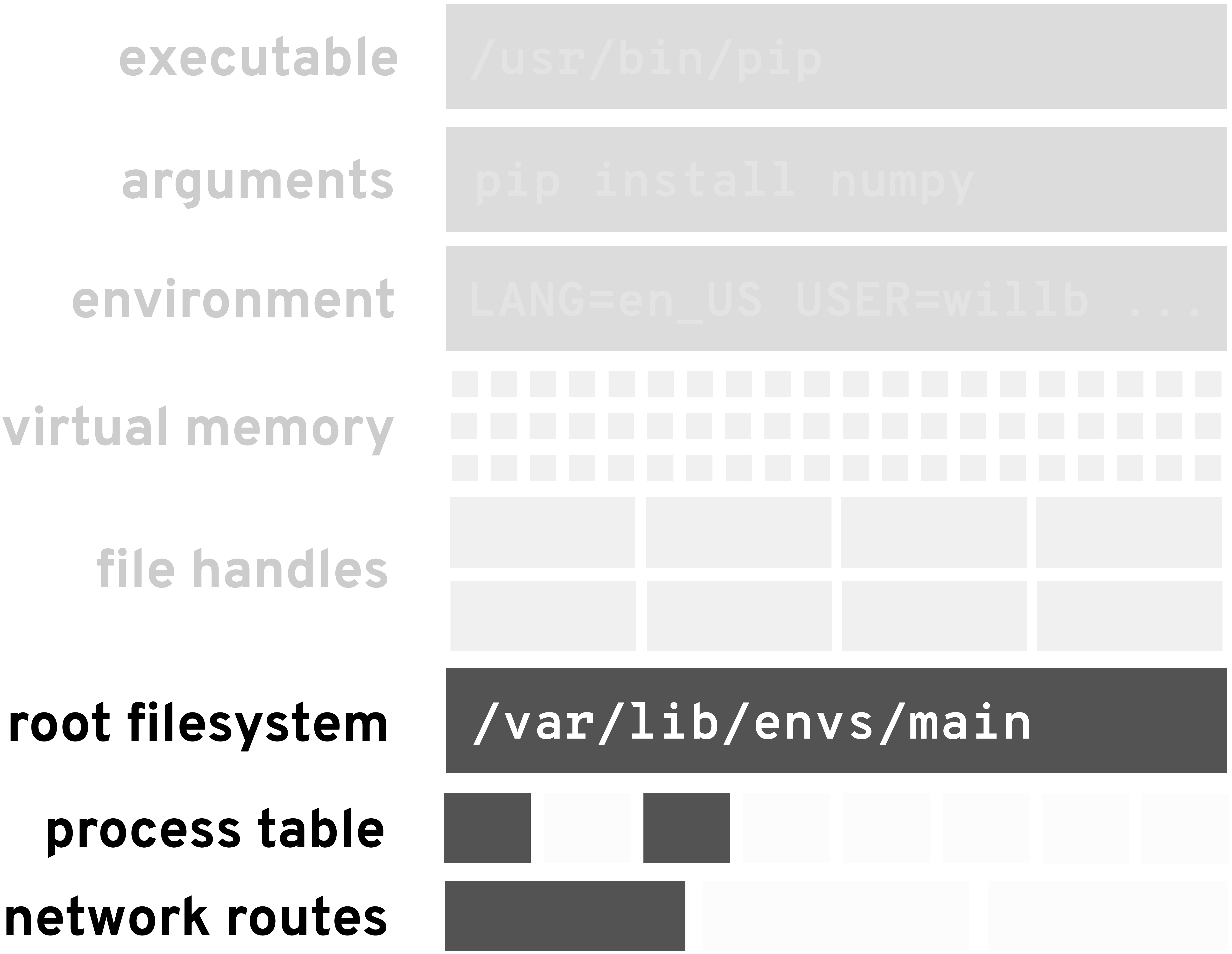
file handles

root filesystem

/

process table

network routes



executable

```
/usr/bin/pip
```

arguments

```
pip install numpy
```

environment

```
LANG=en_US USER=willb ...
```

virtual memory

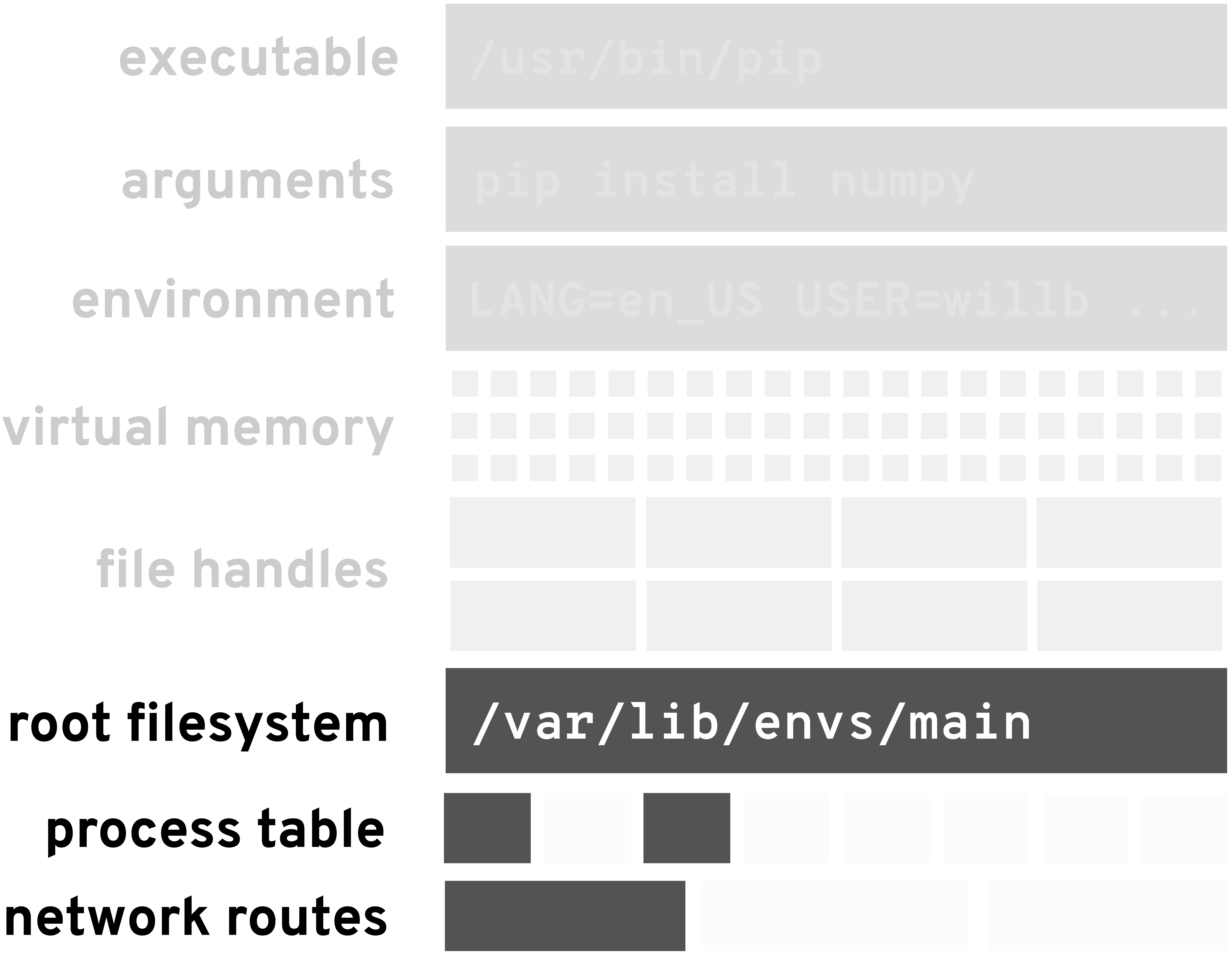
file handles

root filesystem

```
/var/lib/envs/main
```

process table

network routes



executable

/usr/bin/pip

arguments

pip install numpy

environment

LANG=en_US USER=willb ...

virtual memory

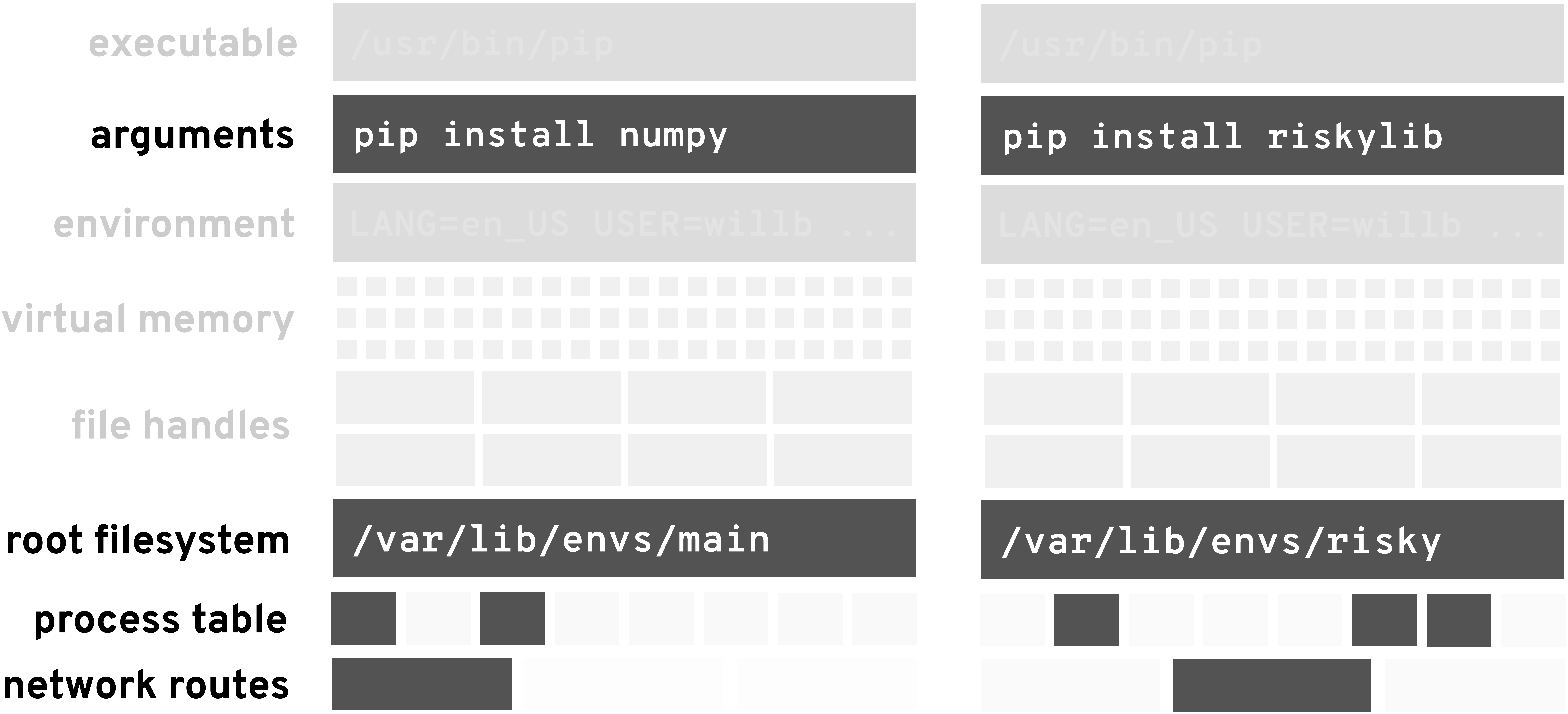
file handles

root filesystem

/var/lib/envs/main

process table

network routes

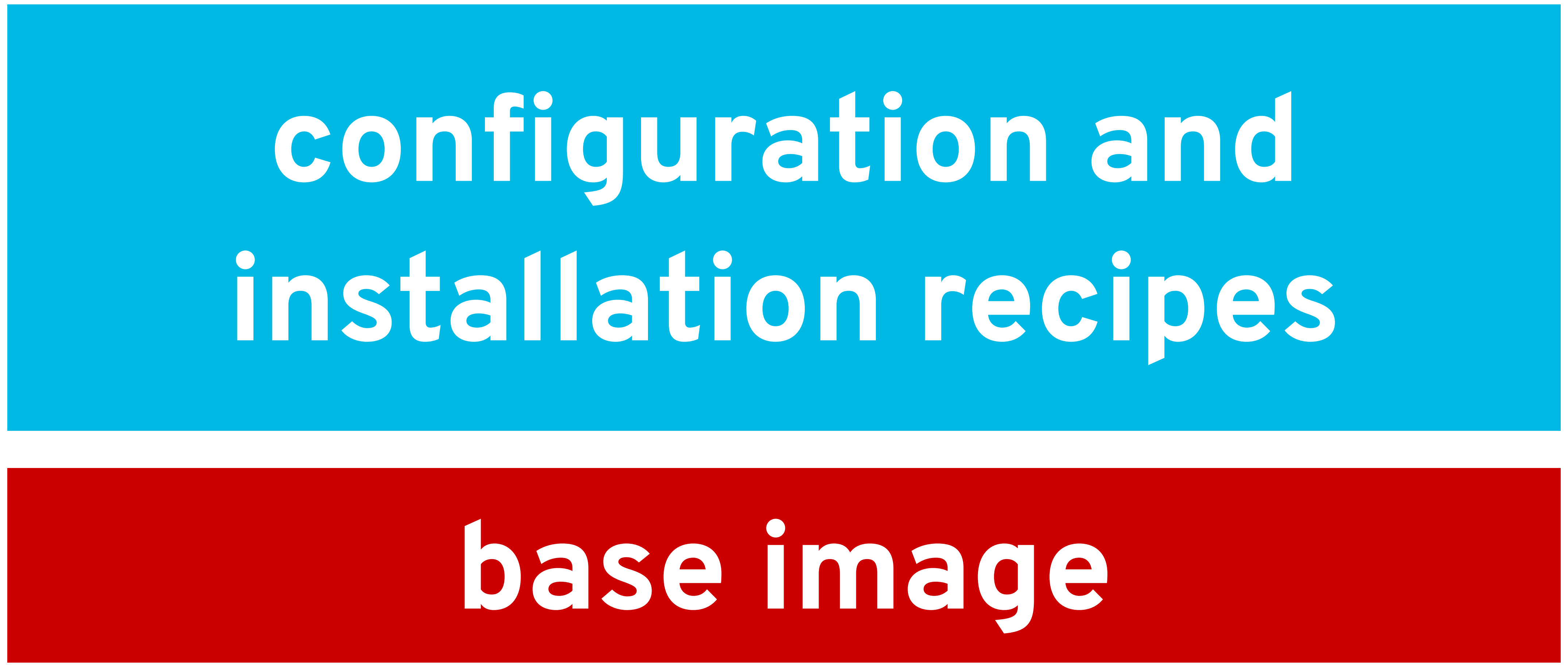


executable	/usr/bin/pip	/usr/bin/pip
arguments	pip install numpy	pip install riskylib
environment	LANG=en_US USER=willb ...	LANG=en_US USER=willb ...
virtual memory		
file handles		
root filesystem	/var/lib/envs/main	/var/lib/envs/risky
process table		
network routes		

Immutable images

base image

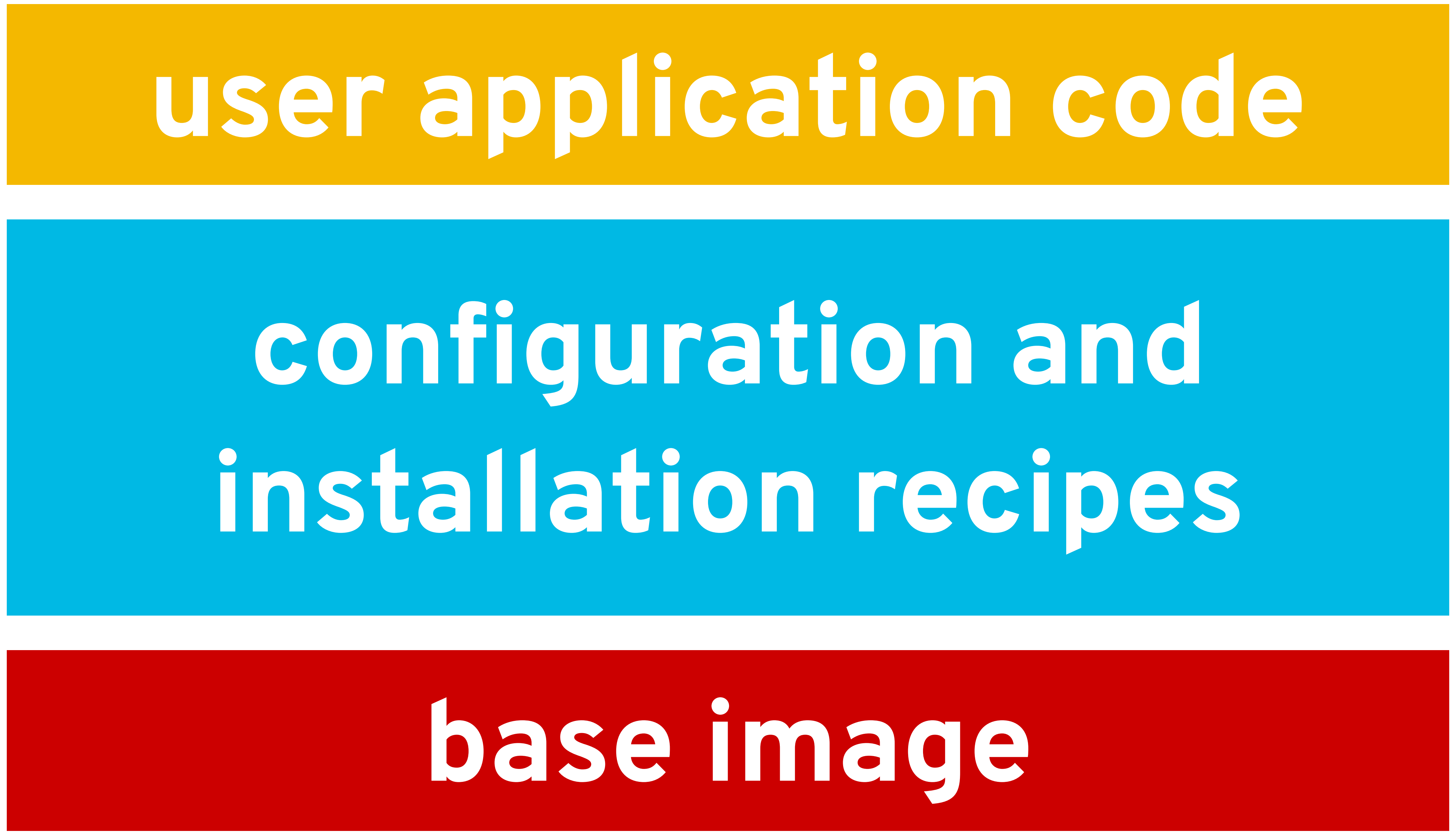
Immutable images



configuration and installation recipes

base image

Immutable images

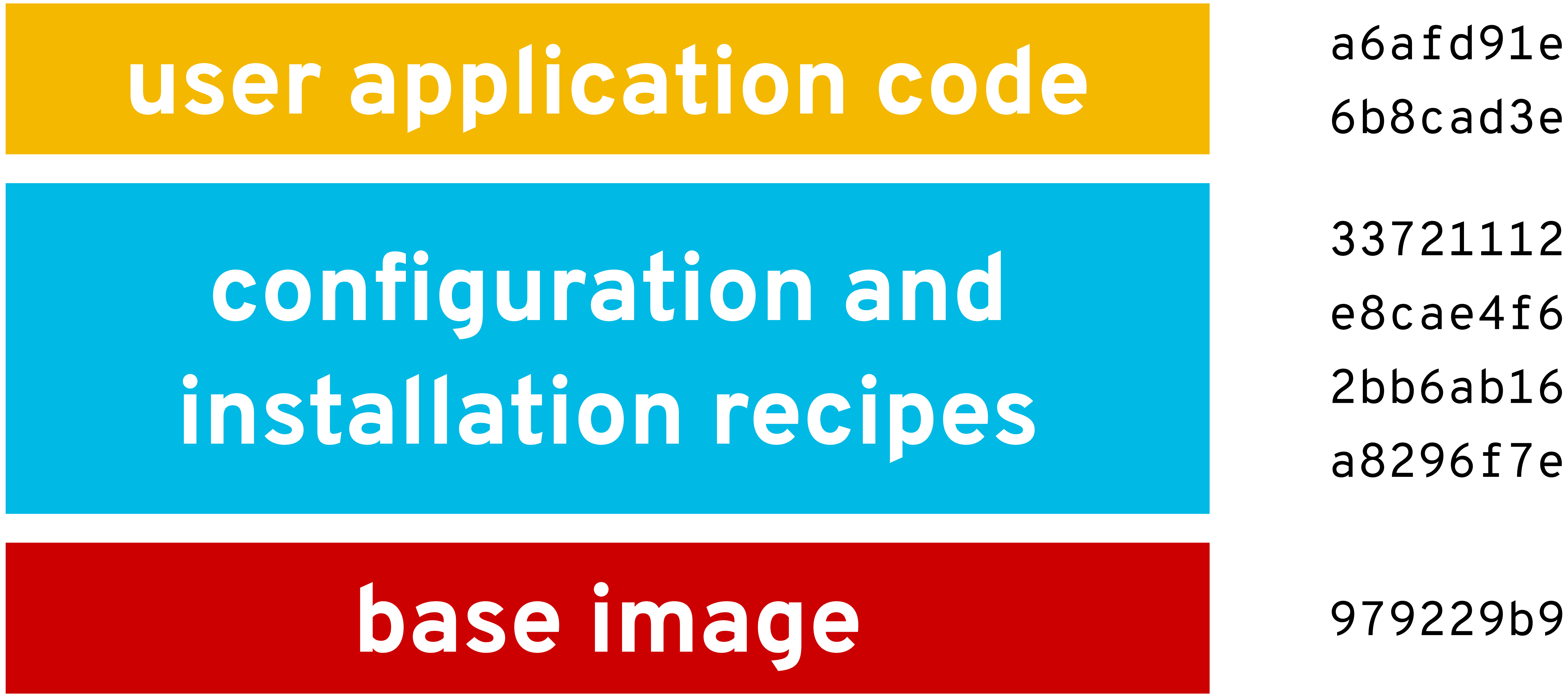


user application code

configuration and installation recipes

base image

Immutable images



user application code

a6afd91e
6b8cad3e

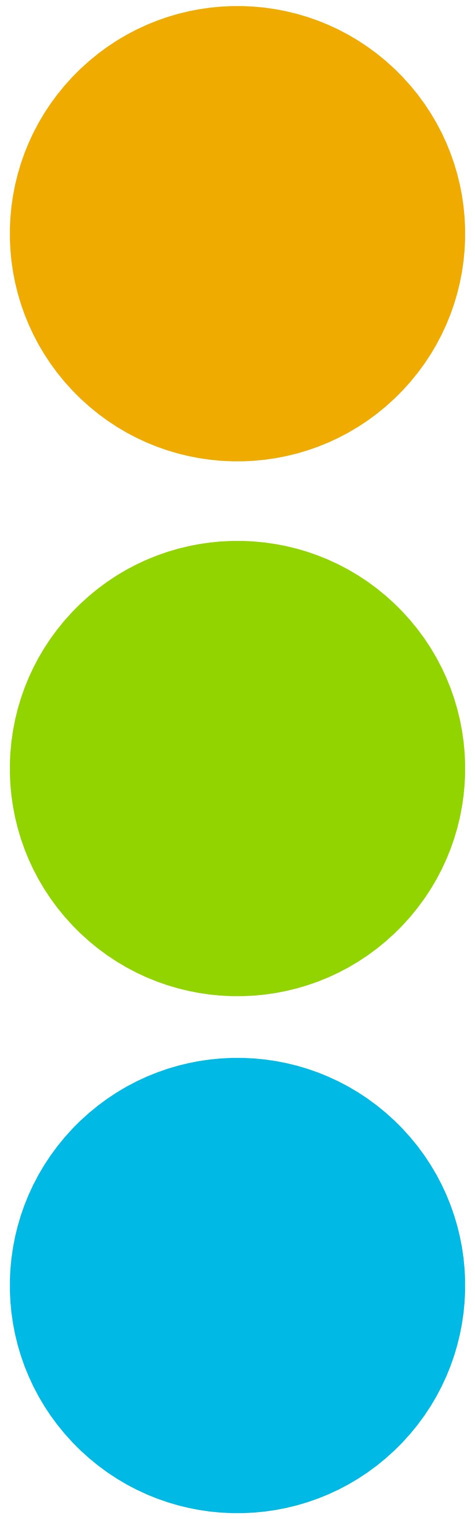
configuration and installation recipes

33721112
e8cae4f6
2bb6ab16
a8296f7e

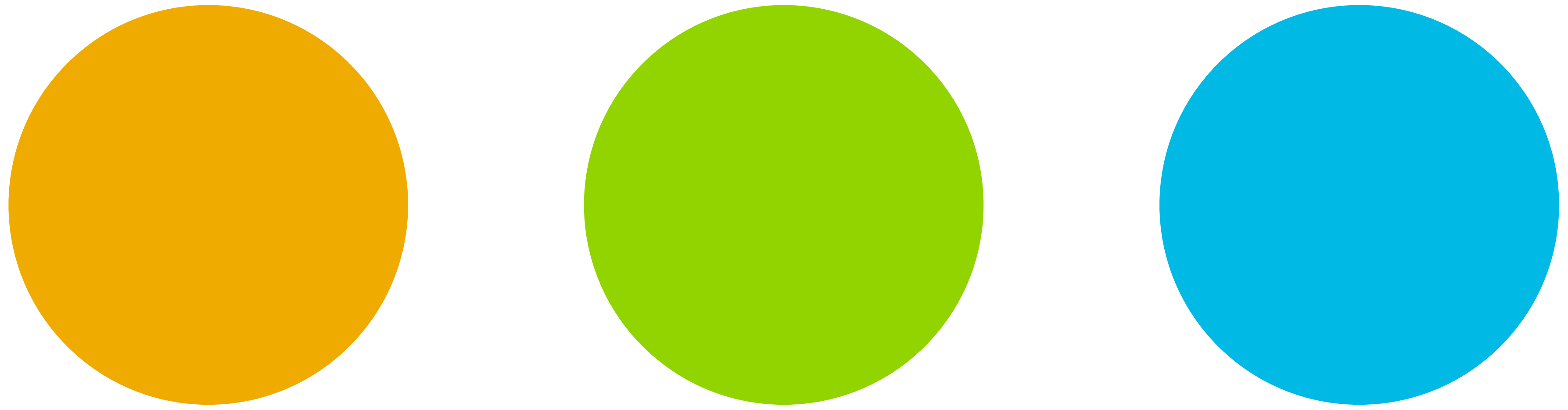
base image

979229b9

Stateless microservices

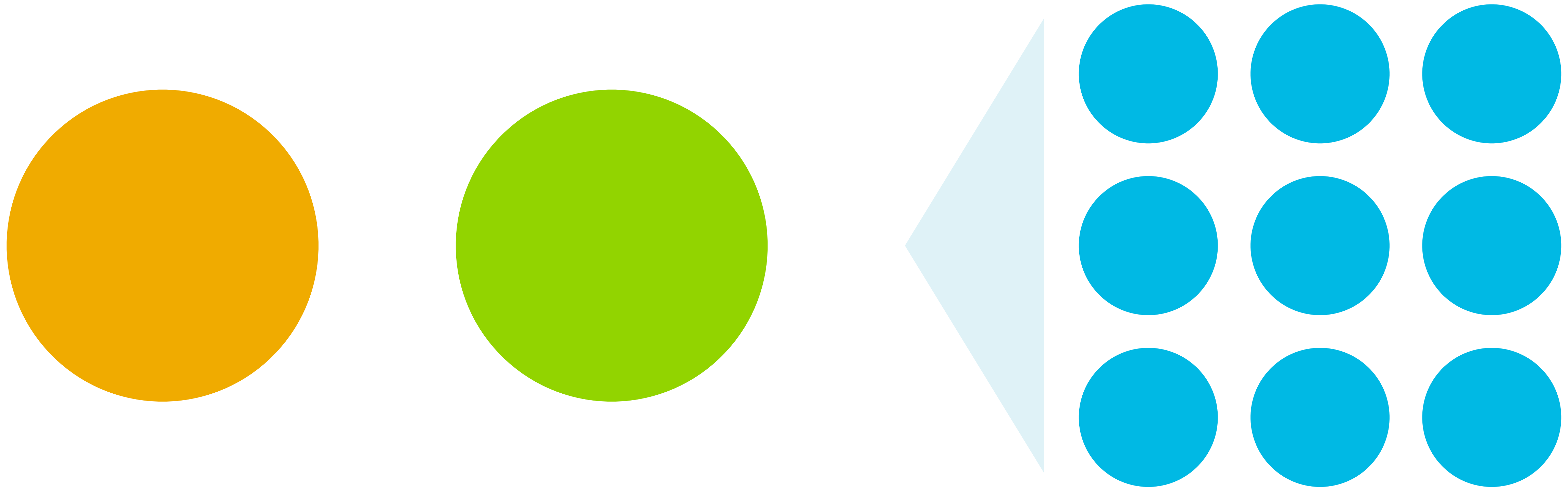


Stateless microservices

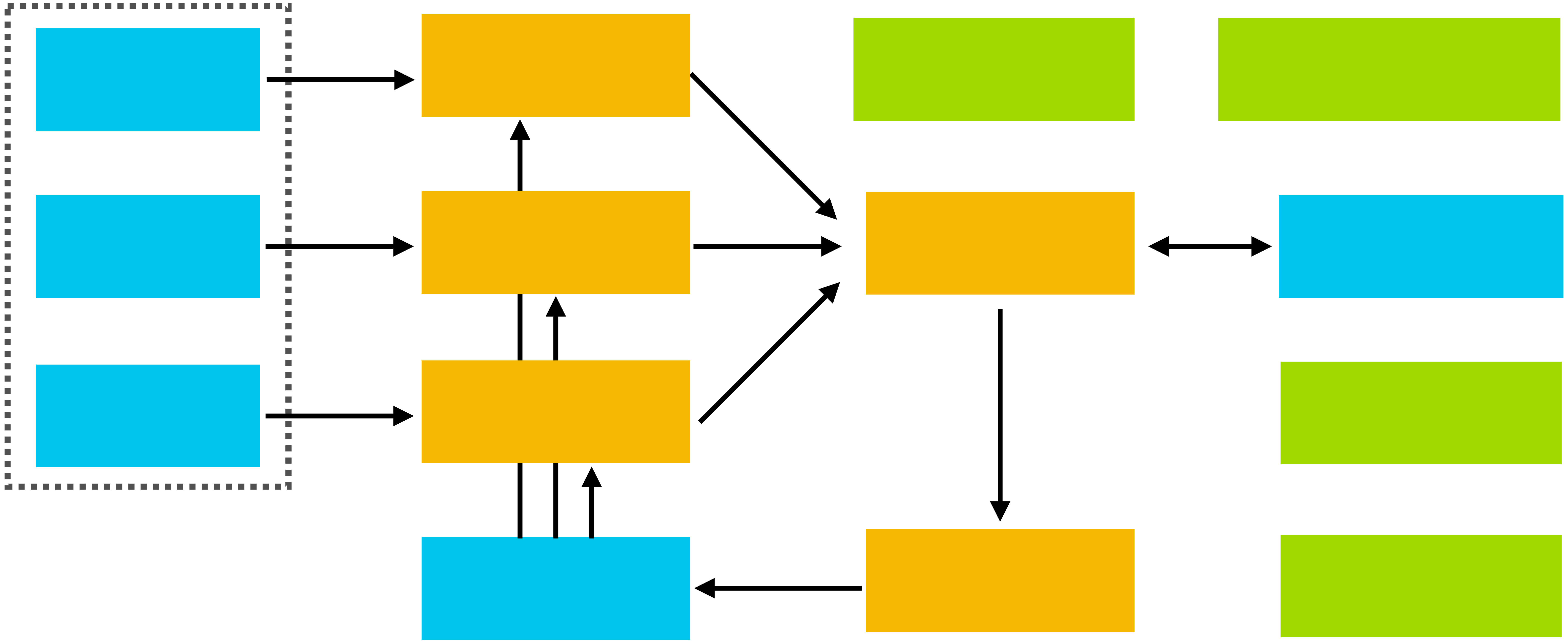


Stateless microservices

Stateless microservices

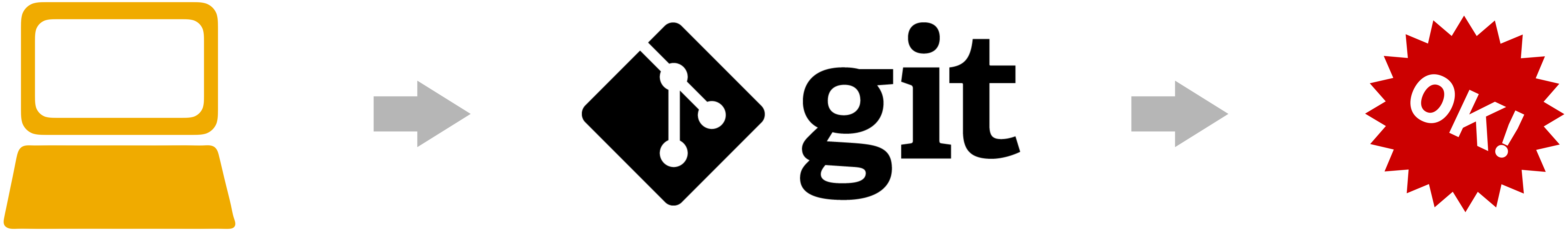


Declarative app configuration

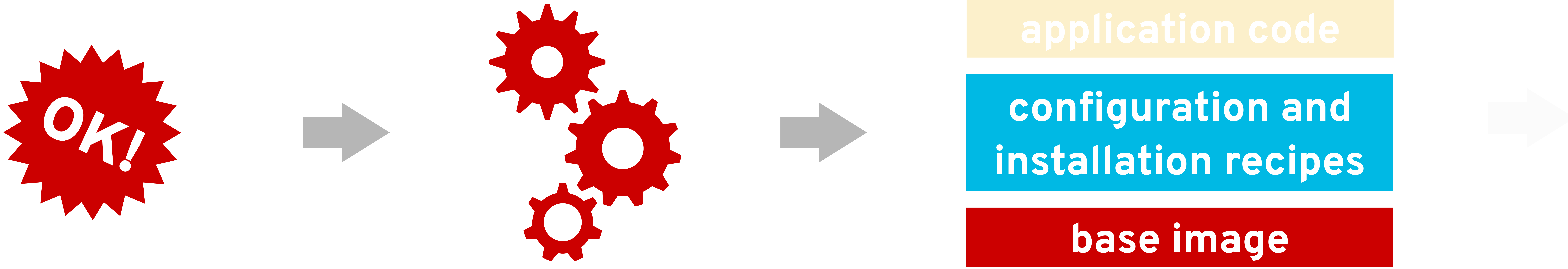


Integration and deployment

Integration and deployment



Integration and deployment

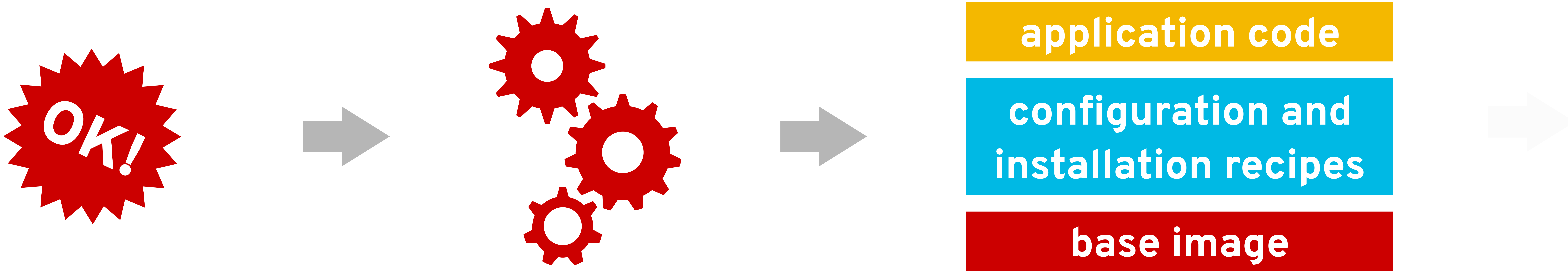


application code

configuration and installation recipes

base image

Integration and deployment



application code

configuration and installation recipes

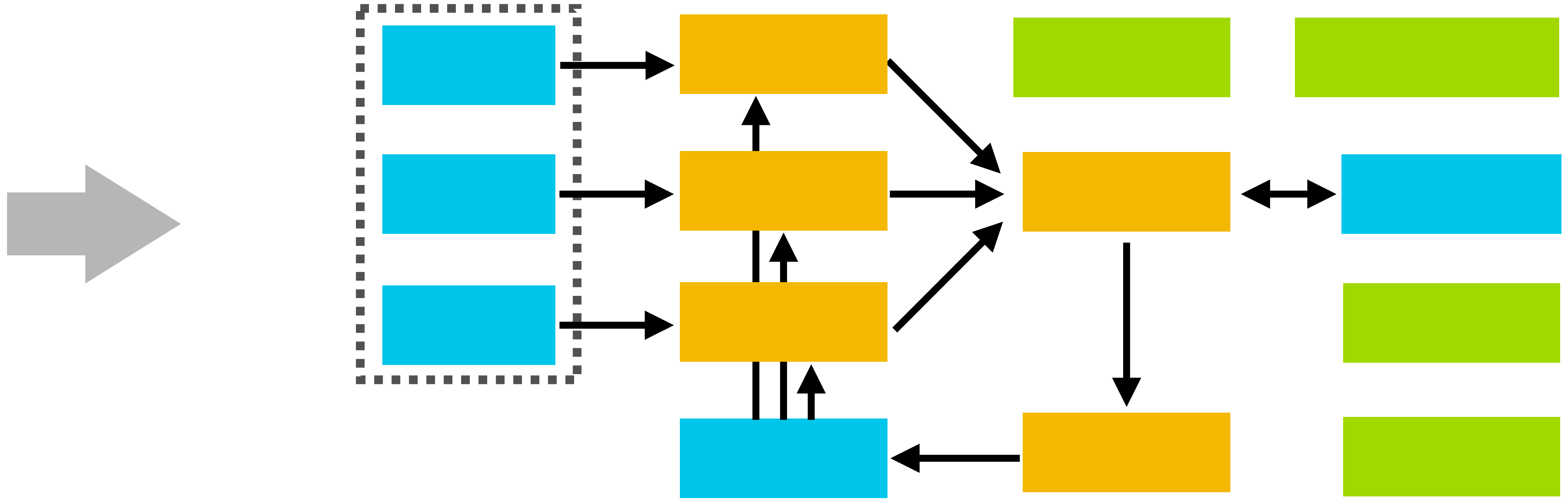
base image

Integration and deployment

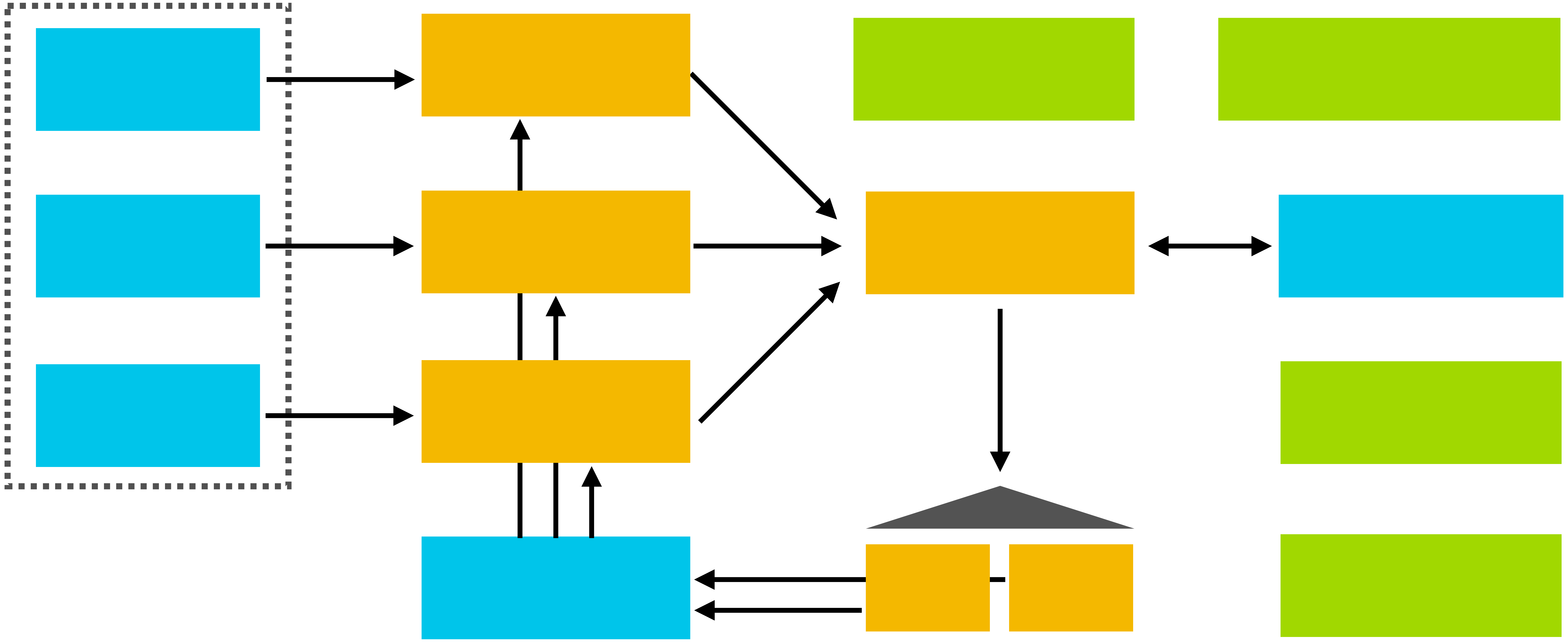
application code

configuration and installation recipes

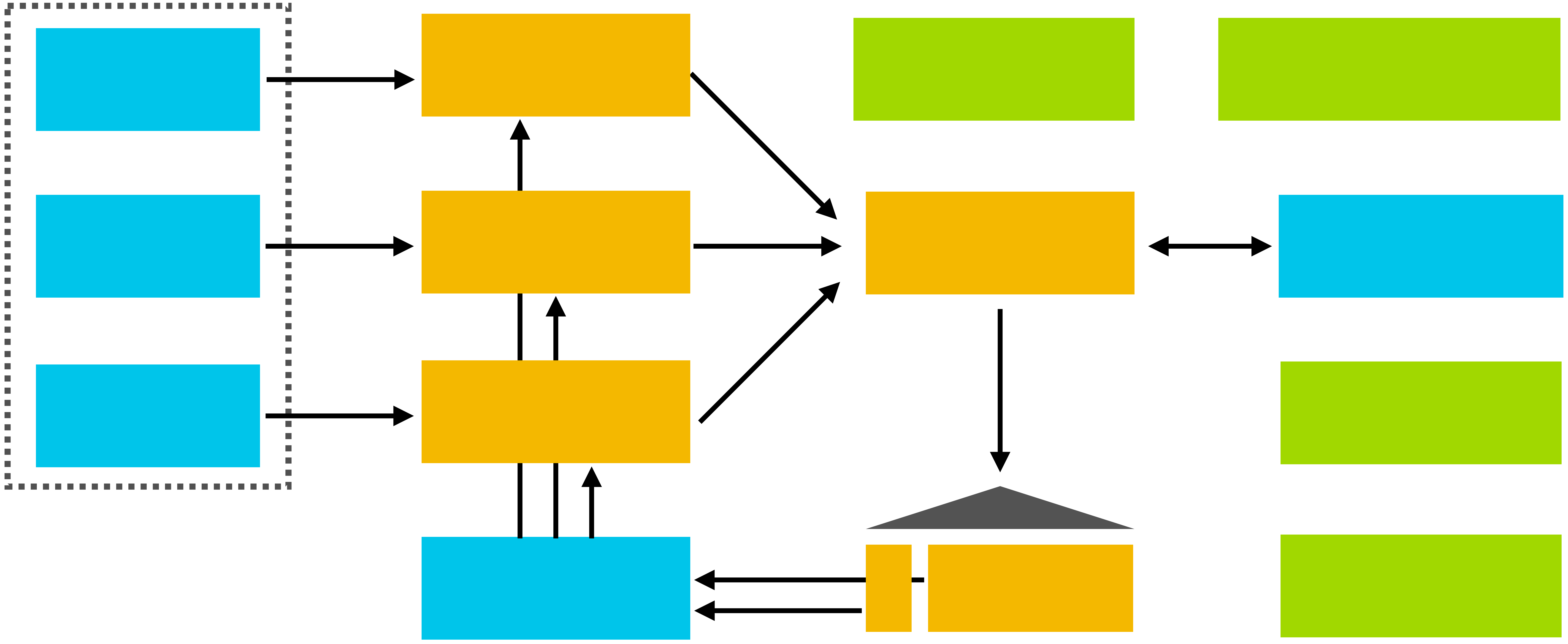
base image



Flexible service routing



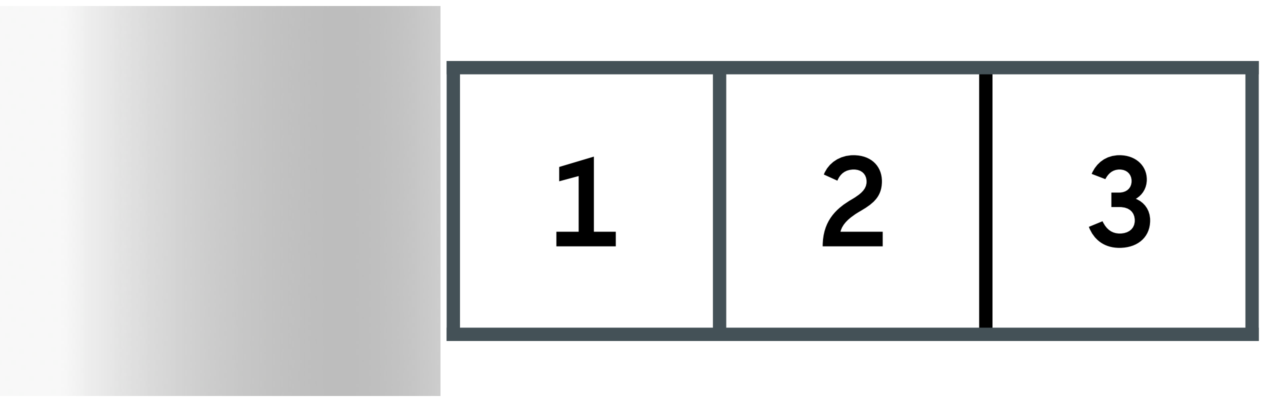
Flexible service routing



How can we build *intelligent applications* in containers?

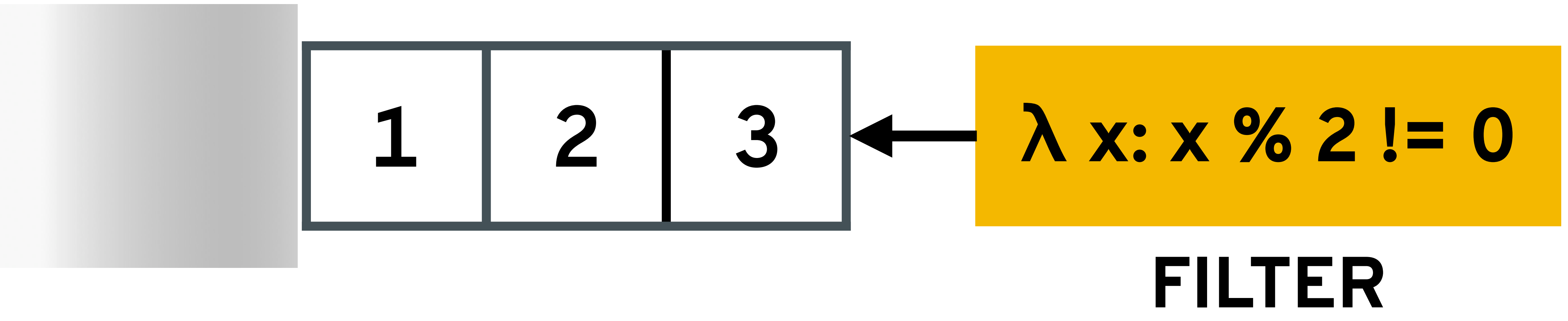
Apache Spark

Apache Spark

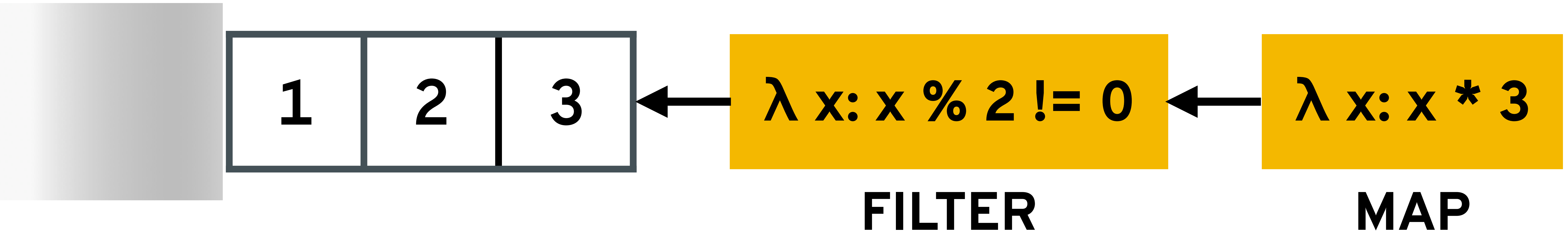


1	2	3

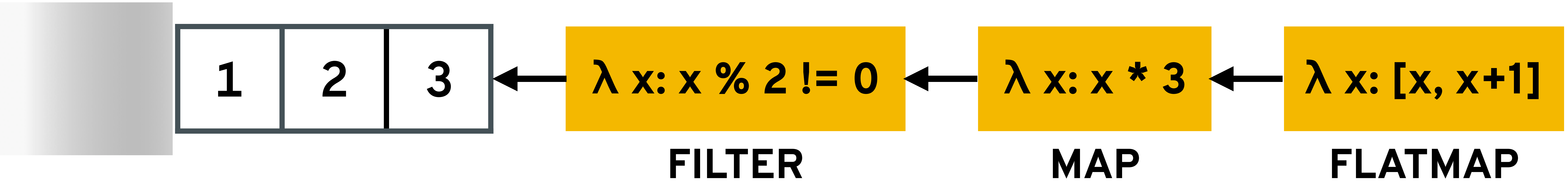
Apache Spark

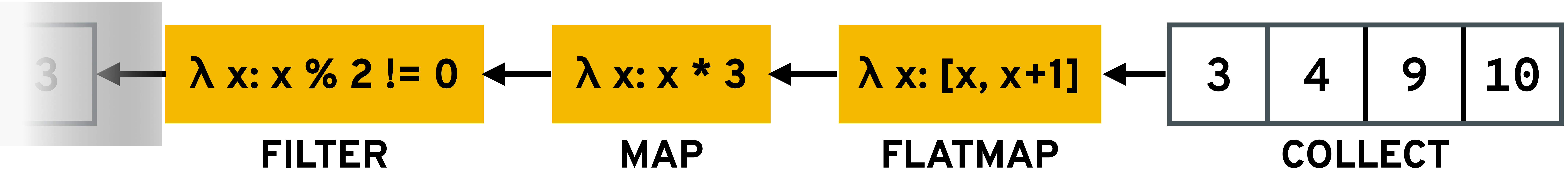


Apache Spark



Apache Spark





3

λ x: x % 2 != 0

FILTER

λ x: x * 3

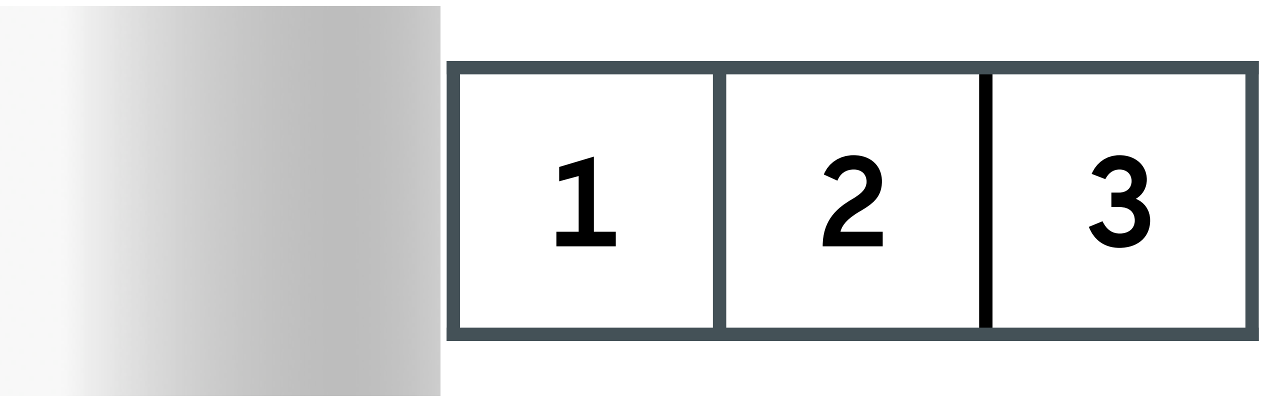
MAP

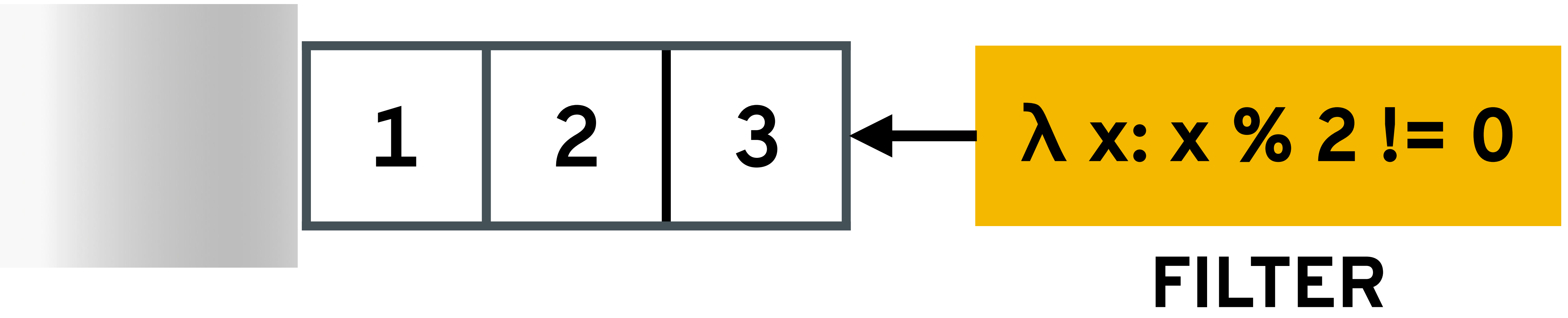
λ x: [x, x+1]

FLATMAP

3	4	9	10

COLLECT

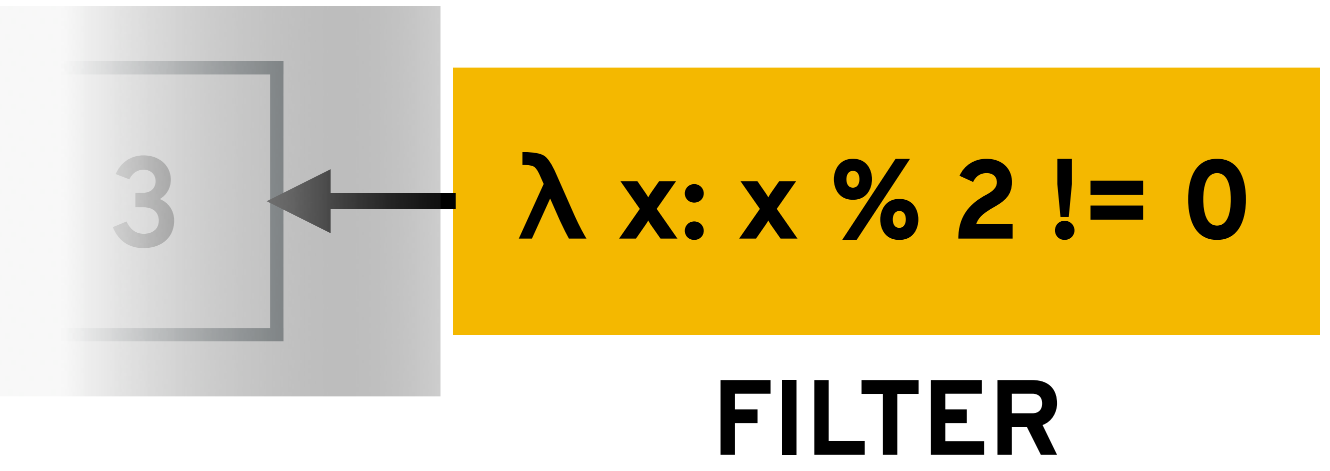

1
2
3



1 2 3

λ x: x % 2 != 0

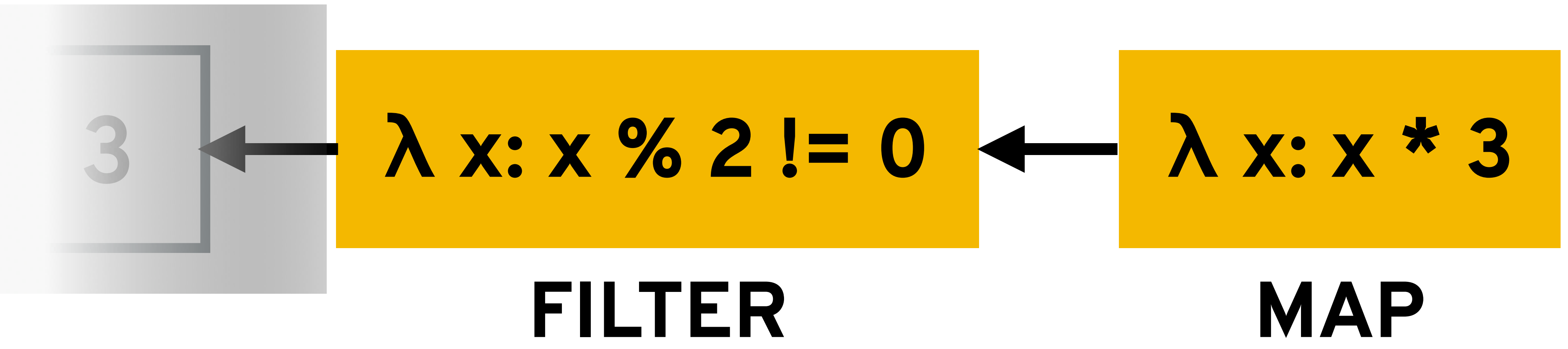
FILTER

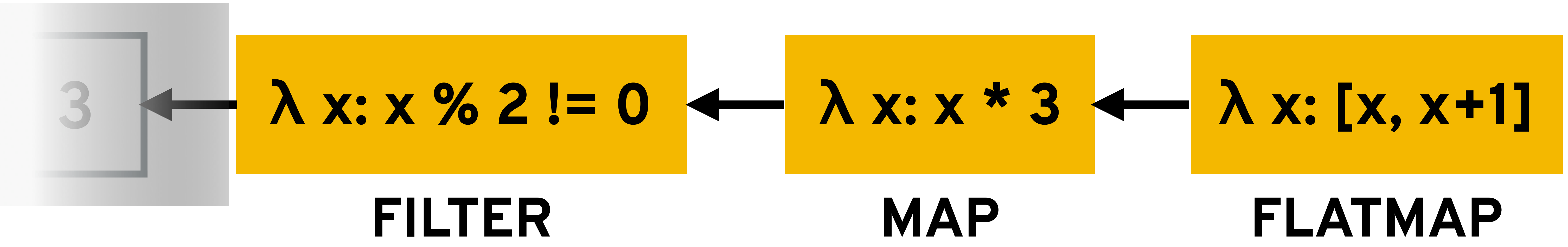


3

λ x: x % 2 != 0

FILTER

3
λ x: x % 2 != 0
FILTER
λ x: x * 3
MAP



3

λ x: x % 2 != 0

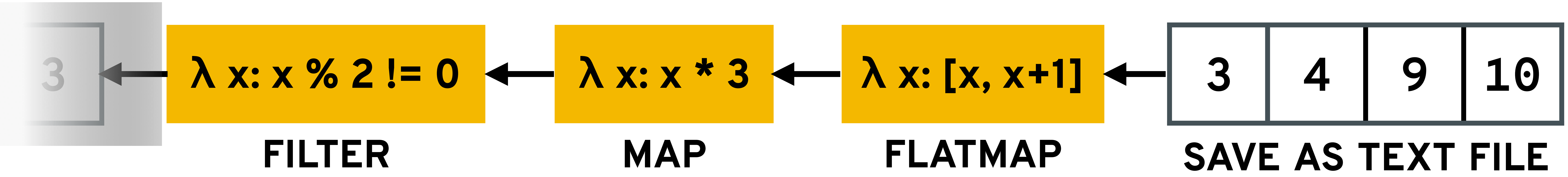
FILTER

λ x: x * 3

MAP

λ x: [x, x+1]

FLATMAP



3

λ x: x % 2 != 0

FILTER

λ x: x * 3

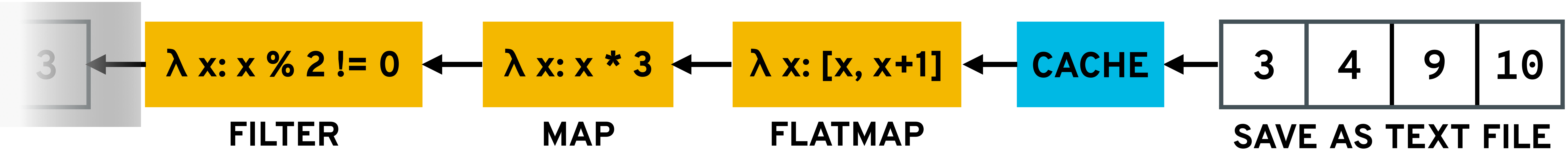
MAP

λ x: [x, x+1]

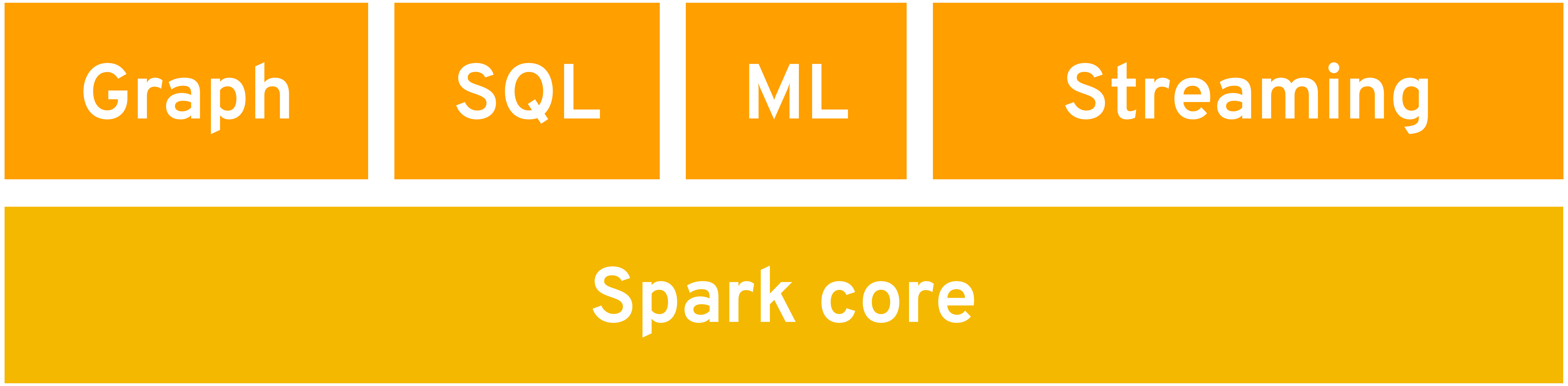
FLATMAP

3	4	9	10

SAVE AS TEXT FILE


3
λ x: x % 2 != 0
FILTER
λ x: x * 3
MAP
λ x: [x, x+1]
FLATMAP
CACHE
3
4
9
10
SAVE AS TEXT FILE

Spark core



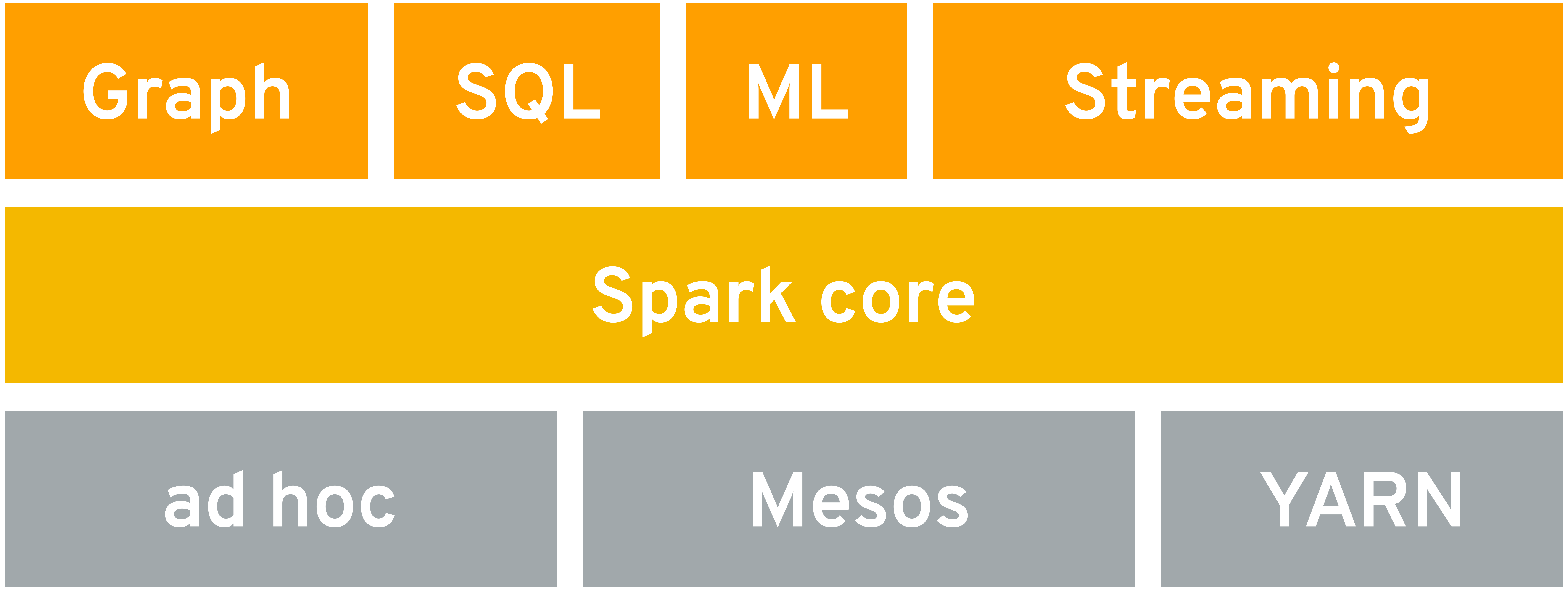
Graph

SQL

ML

Streaming

Spark core



Graph

SQL

ML

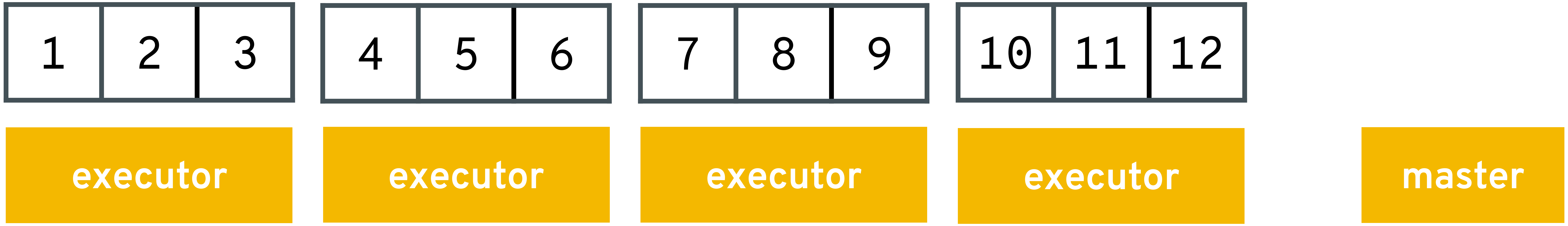
Streaming

Spark core

ad hoc

Mesos

YARN



1 2 3

executor

4 5 6

executor

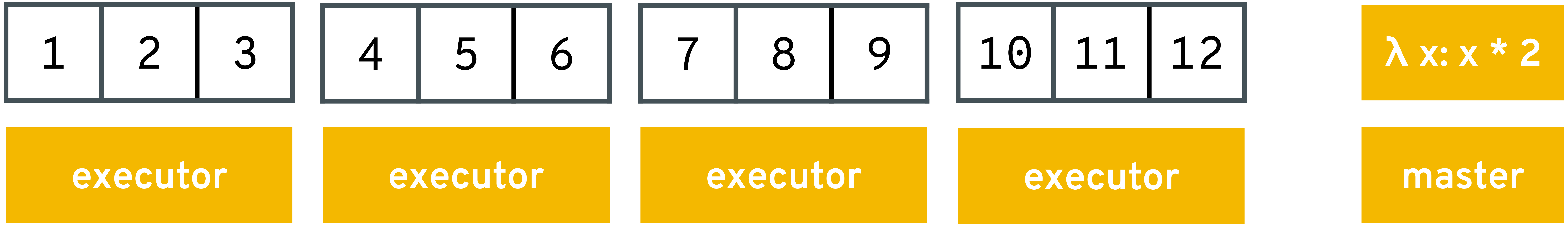
7 8 9

executor

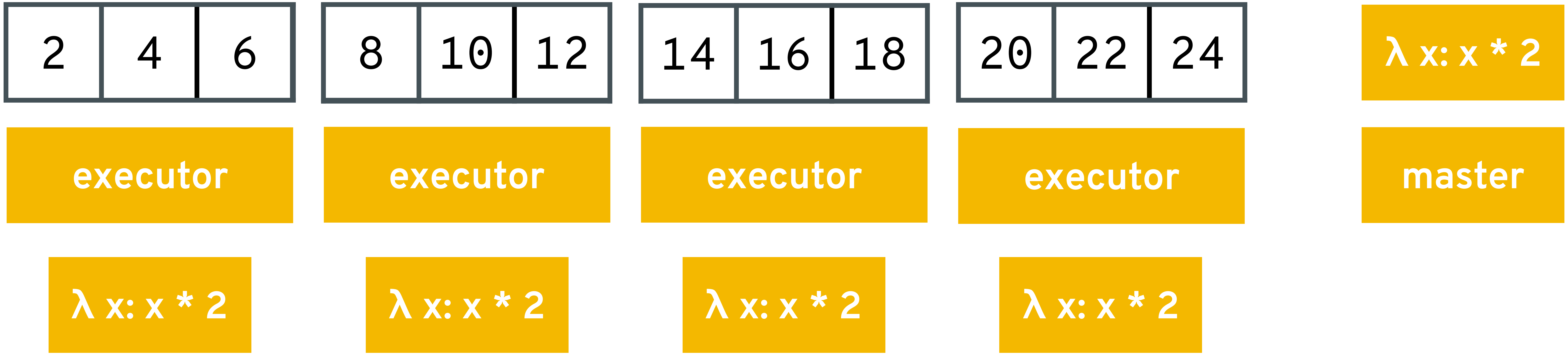
10 11 12

executor

master



1	2	3	4	5	6	7	8	9	10	11	12	λ x: x * 2



2	4	6

8	10	12

14	16	18

20	22	24

λ x: x * 2

executor

executor

executor

executor

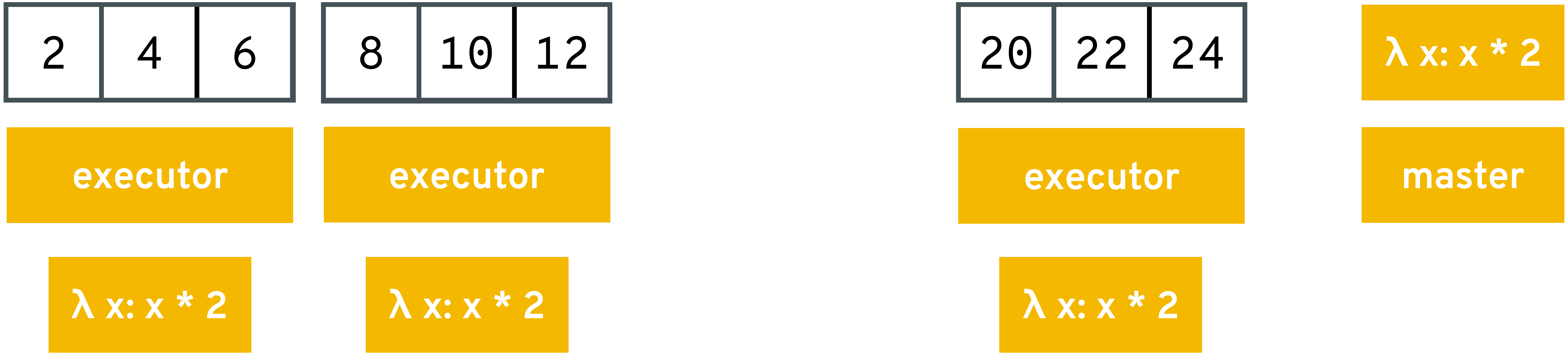
master

λ x: x * 2

λ x: x * 2

λ x: x * 2

λ x: x * 2



2 4 6

8 10 12

20 22 24

λ x: x * 2

executor

executor

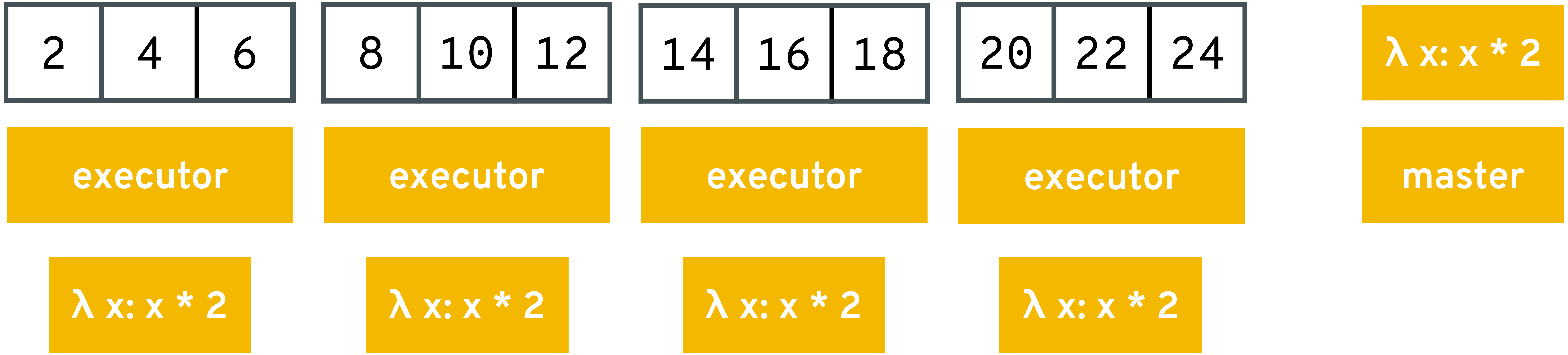
executor

master

λ x: x * 2

λ x: x * 2

λ x: x * 2



2 4 6	8 10 12	14 16 18	20 22 24	λ x: x * 2
executor	executor	executor	executor	master
λ x: x * 2	λ x: x * 2	λ x: x * 2	λ x: x * 2	

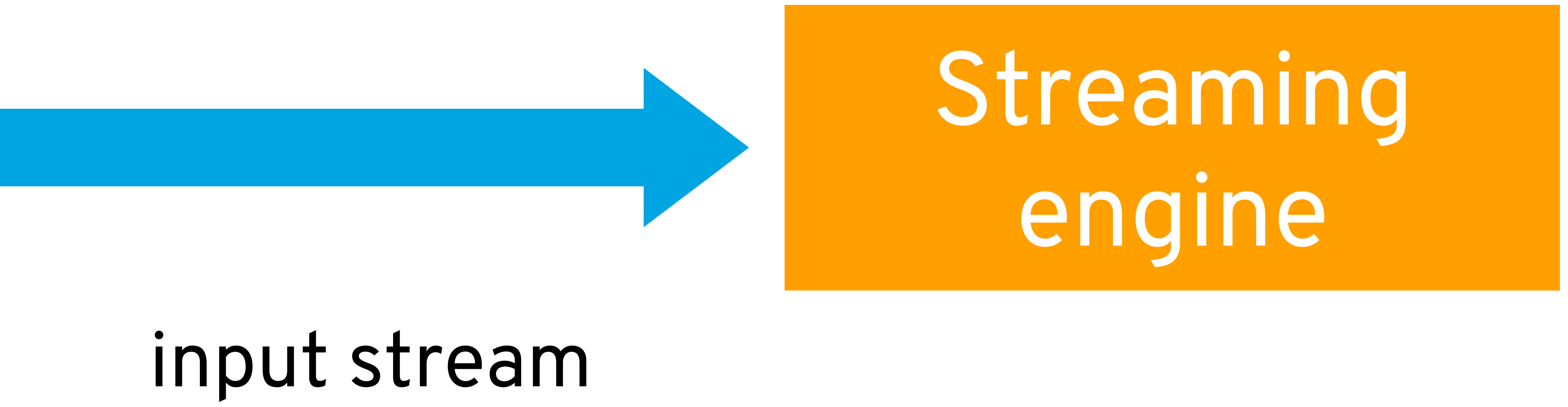
Streaming data

Goal: use the **same abstraction** for batch and "streaming" (micro-batch) data by **dividing a stream into many small RDDs.**

input stream

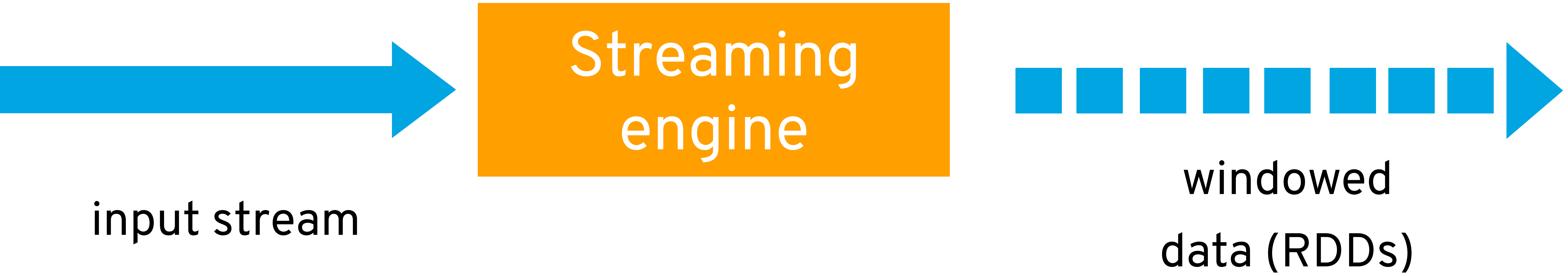
Streaming data

Goal: use the **same abstraction** for batch and "streaming" (micro-batch) data by **dividing a stream into many small RDDs.**



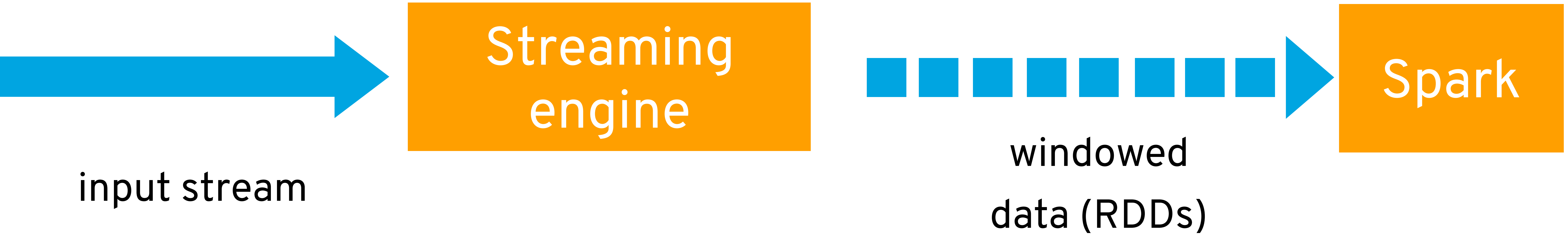
Streaming data

Goal: use the **same abstraction** for batch and “streaming” (micro-batch) data by **dividing a stream into many small RDDs.**



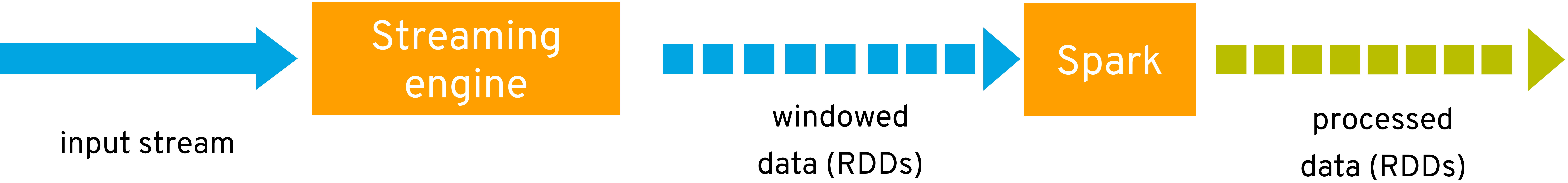
Streaming data

Goal: use the **same abstraction** for batch and "streaming" (micro-batch) data by **dividing a stream into many small RDDs.**



Streaming data

Goal: use the **same abstraction** for batch and "streaming" (micro-batch) data by **dividing a stream into many small RDDs.**



Structured queries

The capacity to run arbitrary code in RDDs is powerful but comes with **an important tradeoff:** Spark can’t rearrange RDD programs to improve their performance.

Writing Spark programs with a **query DSL** allows Spark to generate **optimized execution plans.**

Structured query in Spark

SQL interface (unchecked syntax or semantics)

```
SELECT word, COUNT(*) FROM words GROUP BY word
```

Data frame interface (semantics checked at run-time)

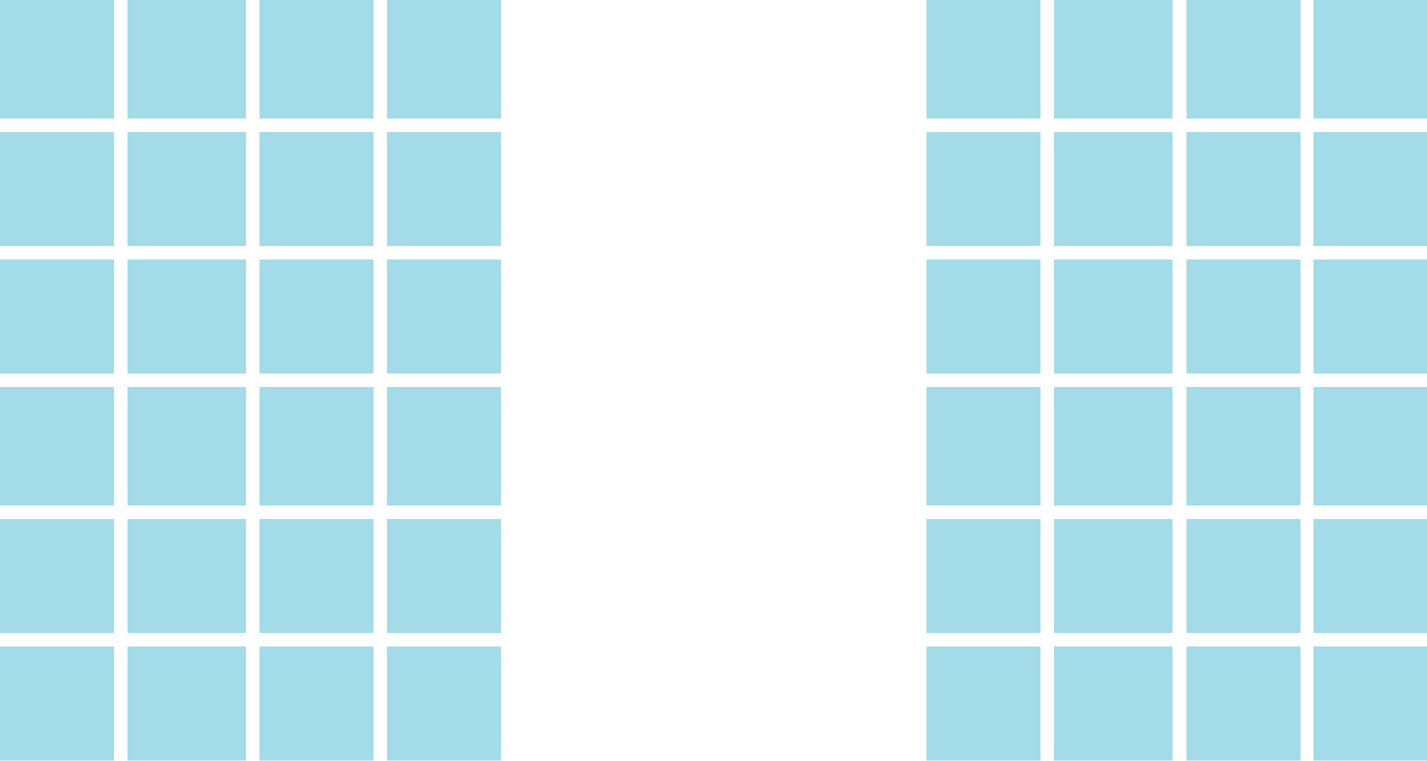
```
words.groupBy('word').count()
```

Dataset interface (mostly checked at compile-time)

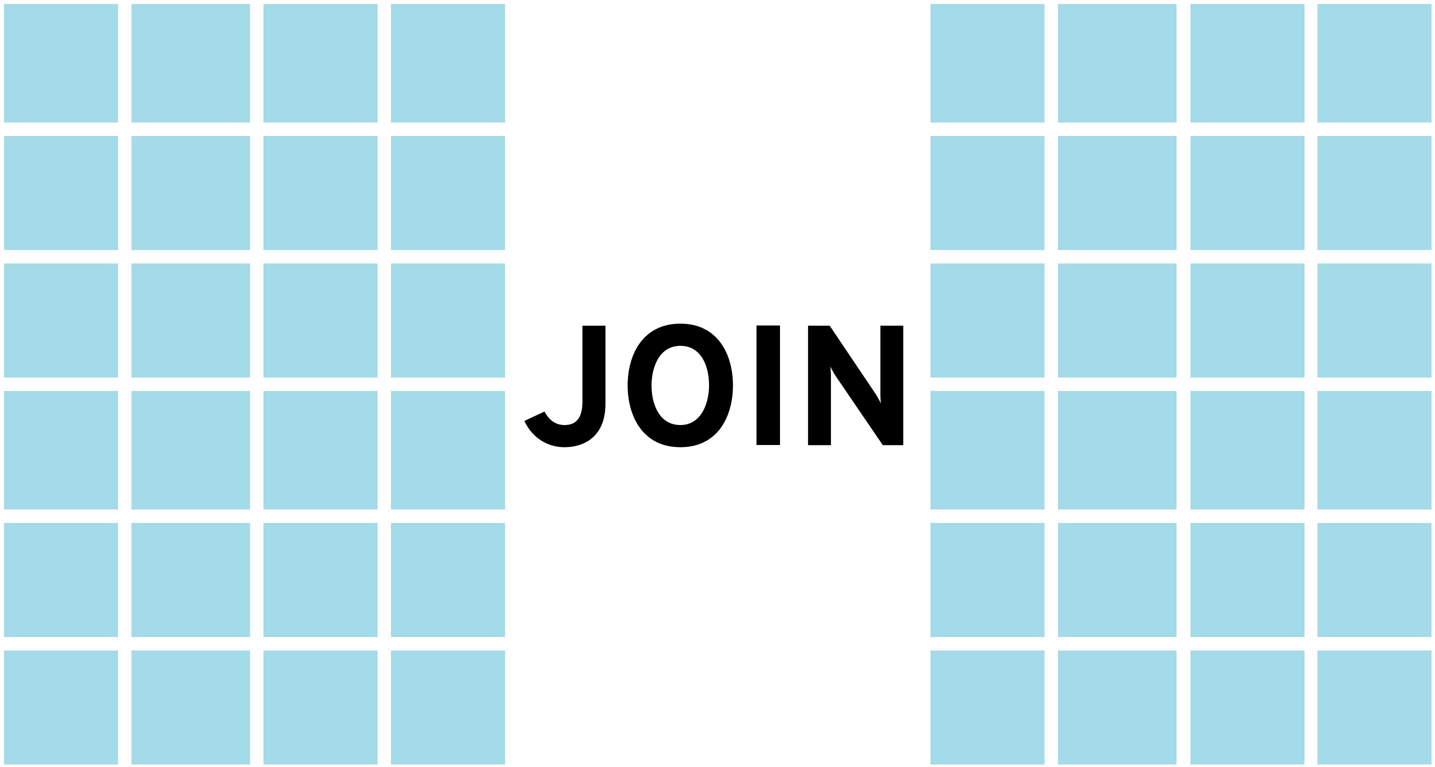
Query planning

```
SELECT * FROM A, B WHERE
  A.ID = B.ID AND
  uncommon(A.X) AND
  extremelyRare(B.Y)
```

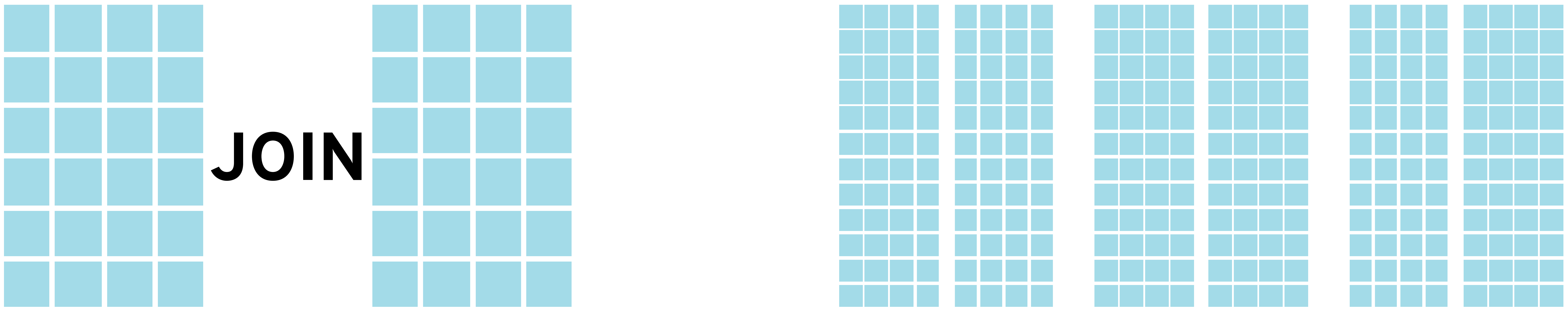
A naïve plan



A naïve plan

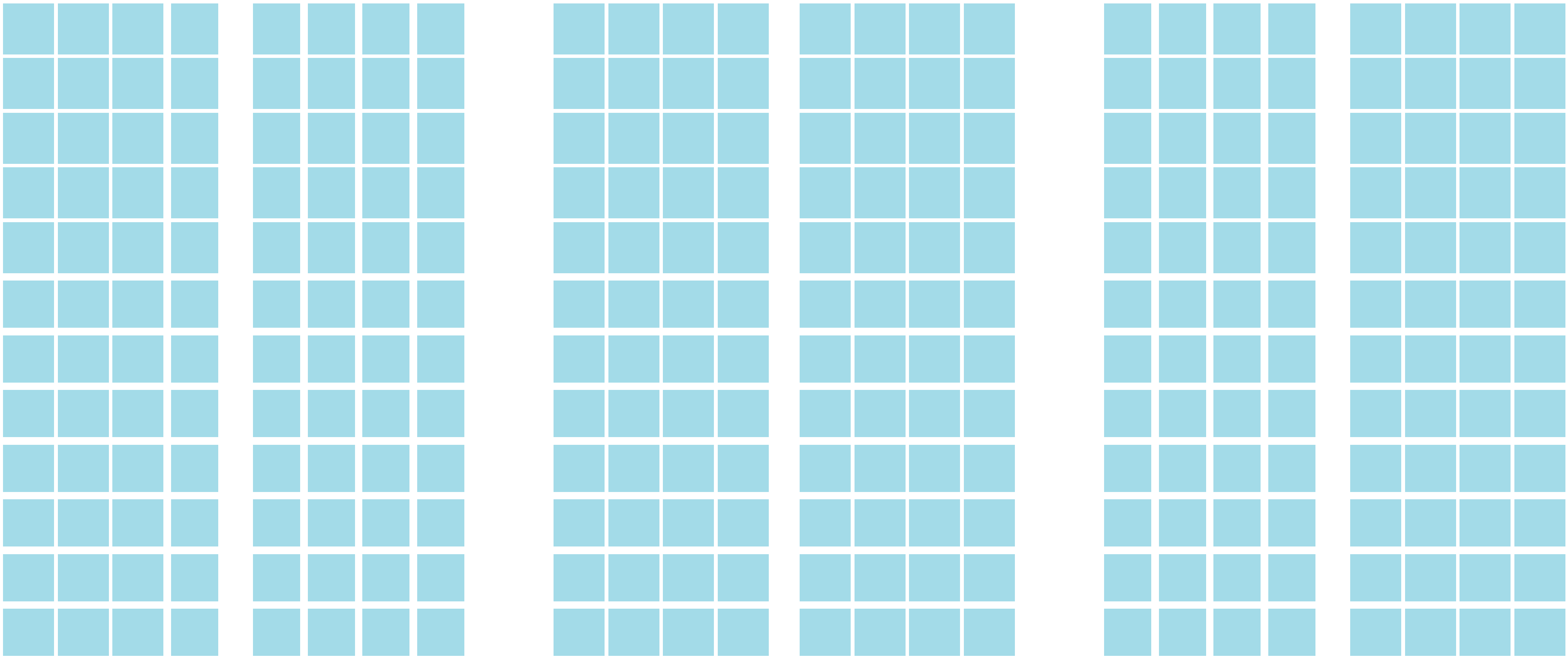


A naïve plan

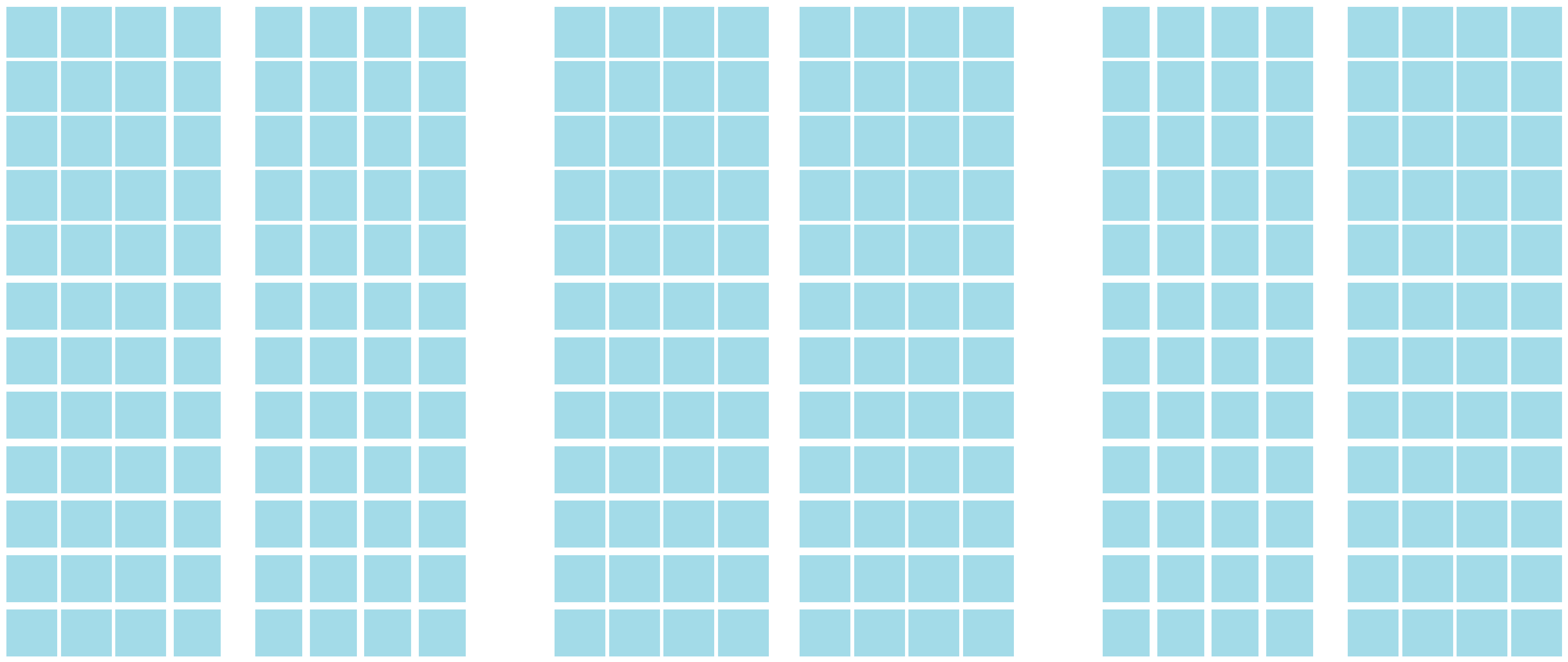


JOIN

A naïve plan

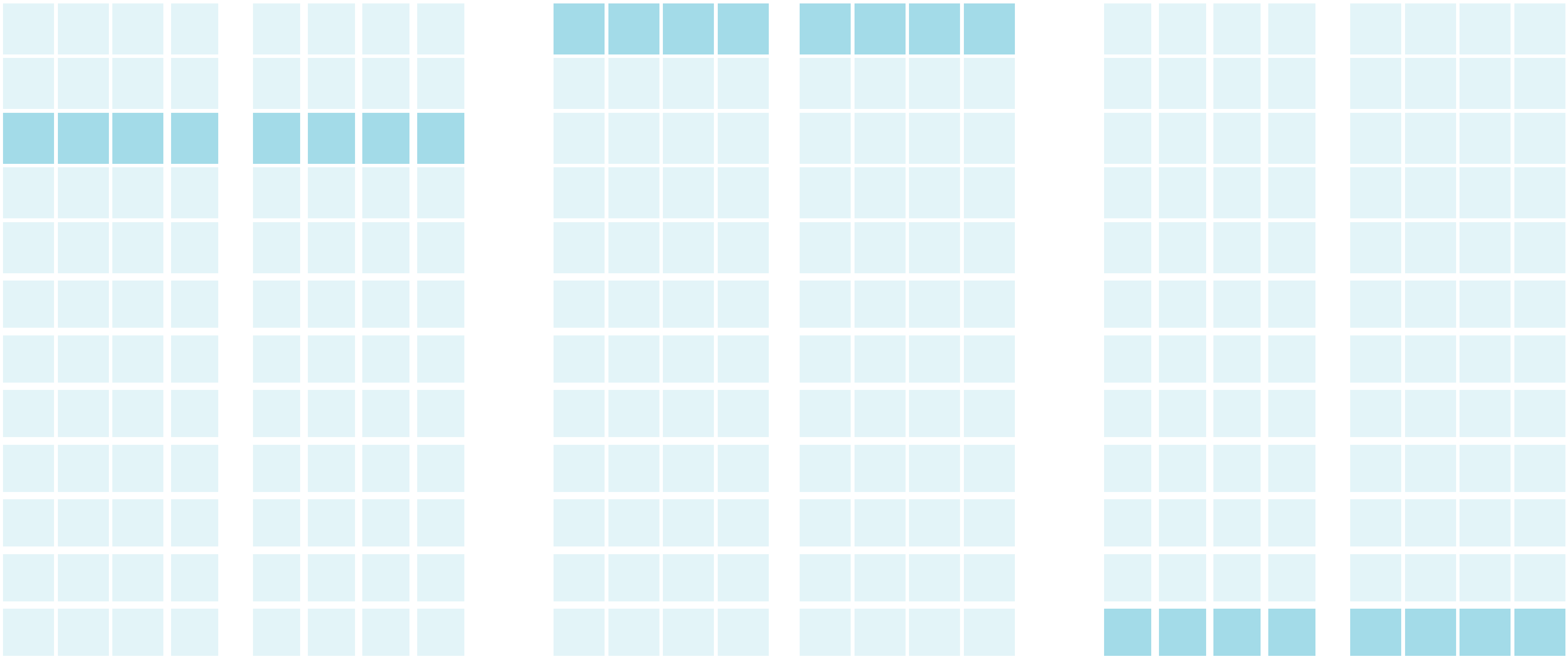


A naïve plan



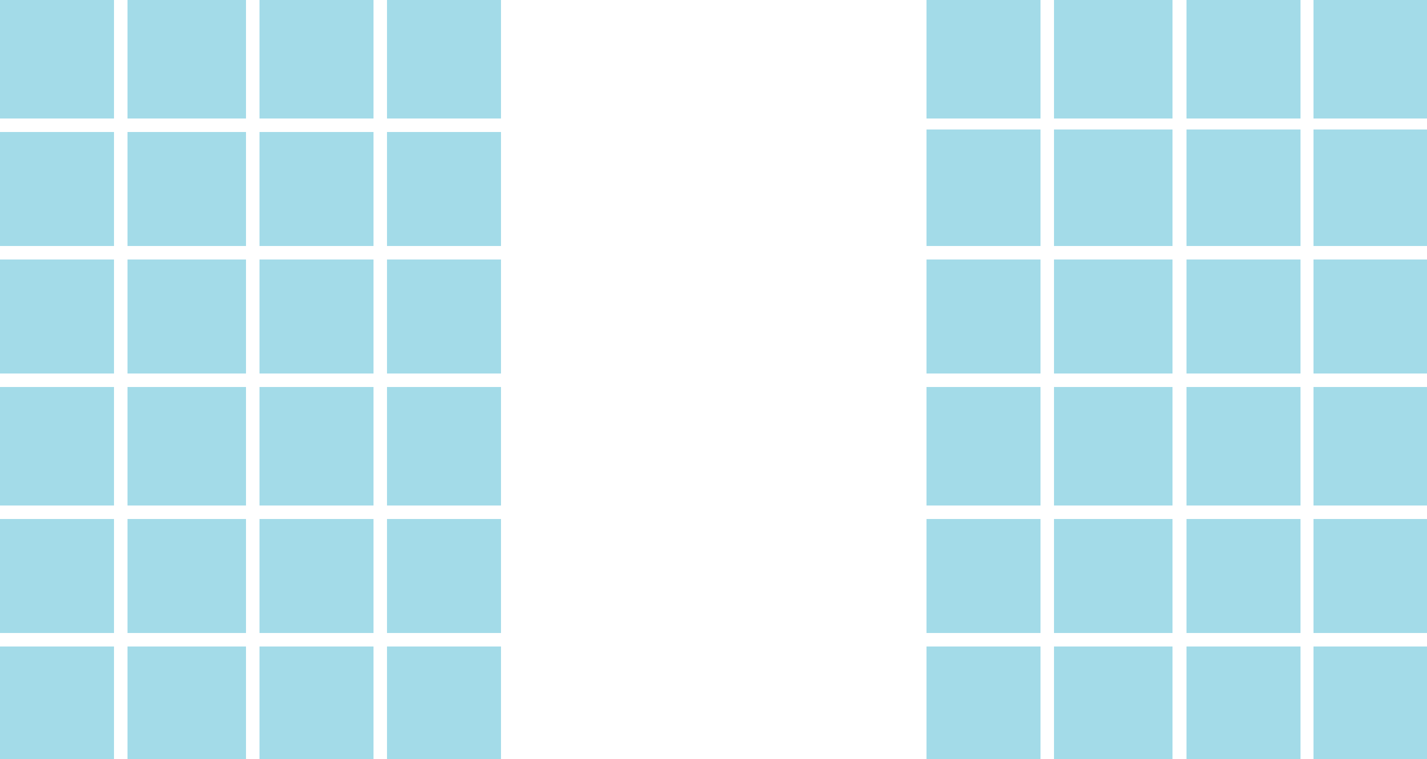
FILTER

A naïve plan

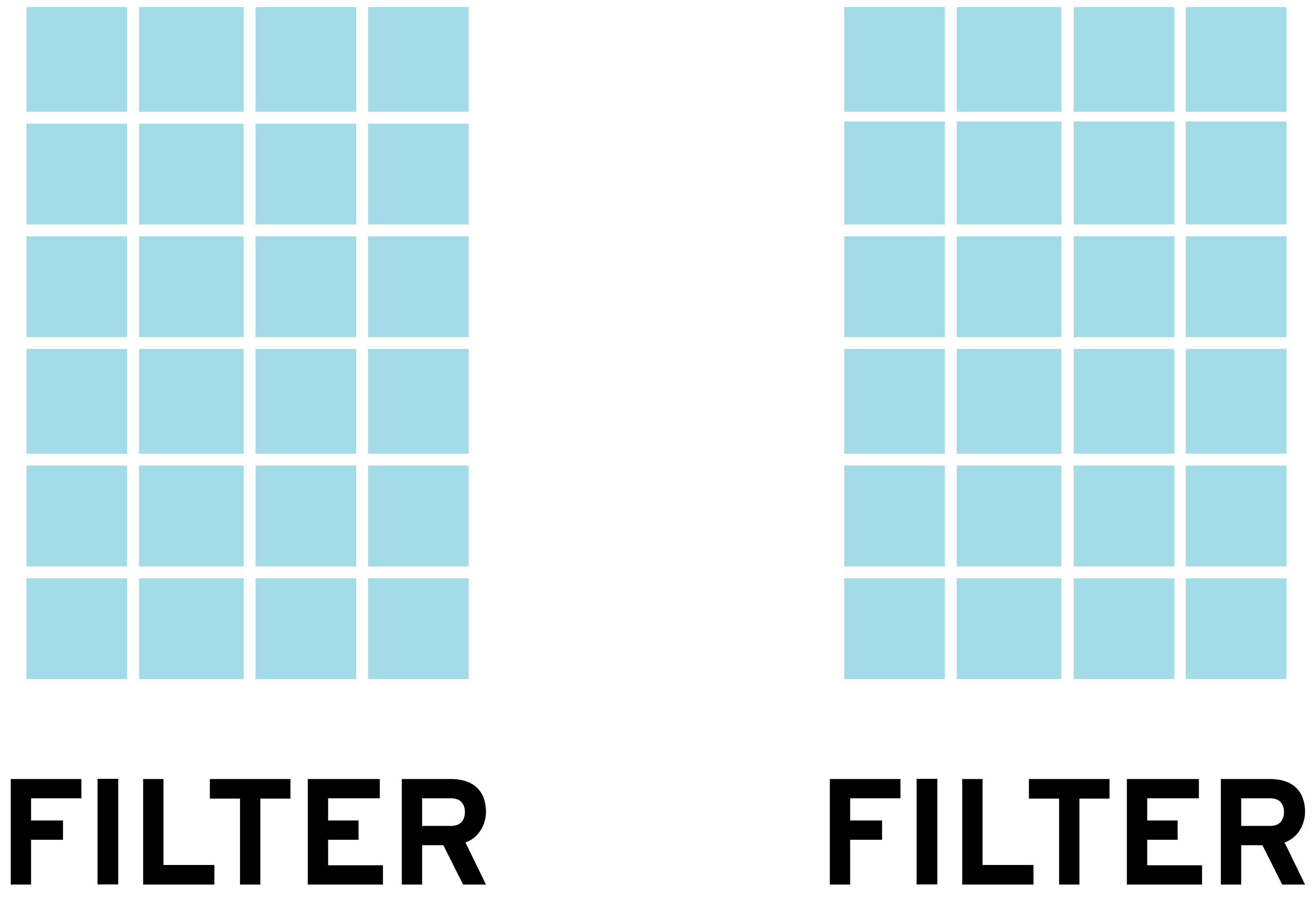


FILTER

An optimized plan



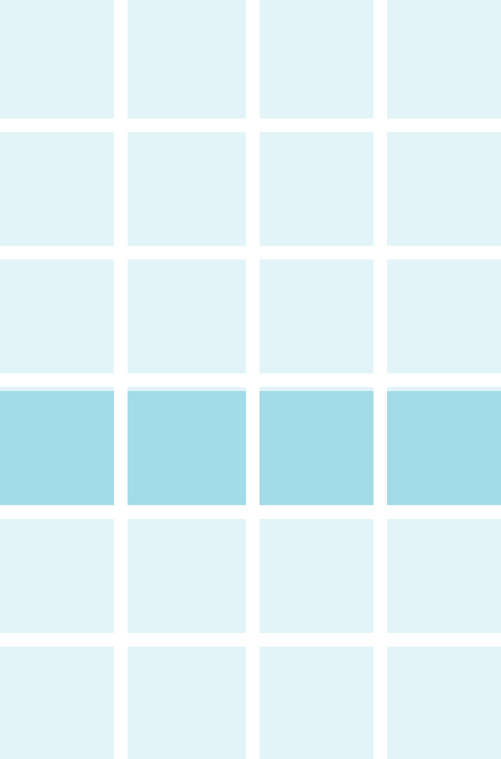
An optimized plan



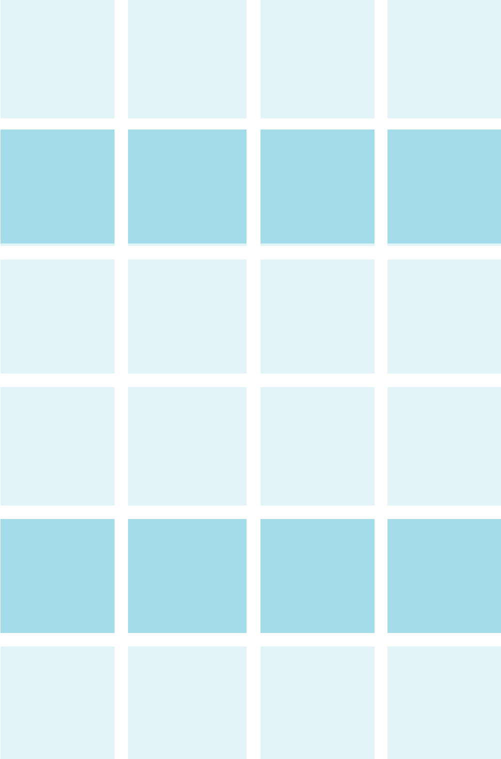
FILTER

FILTER

An optimized plan

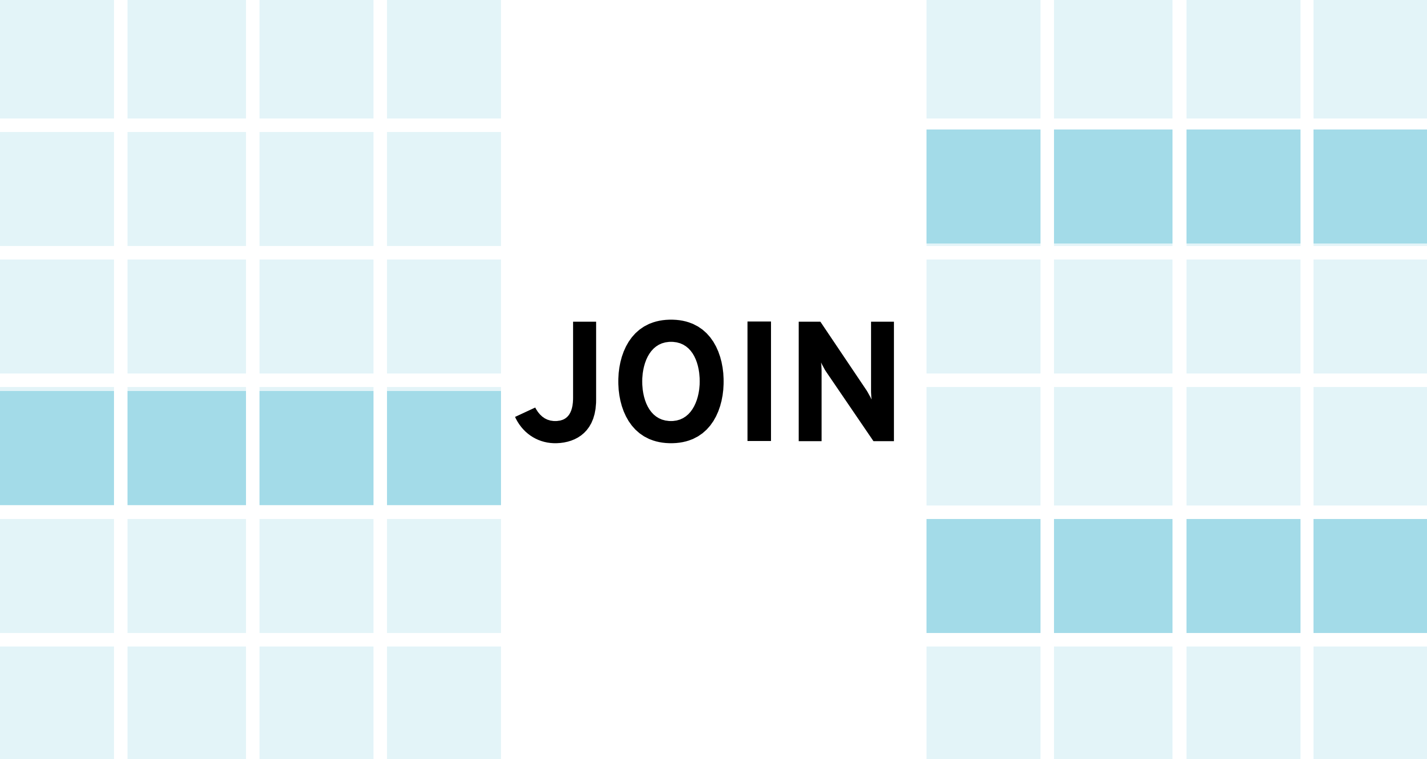


FILTER



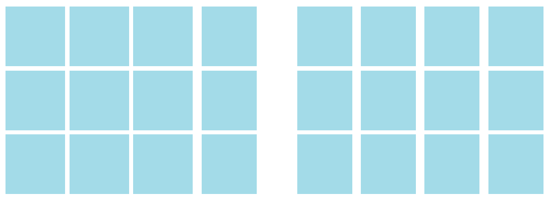
FILTER

An optimized plan



JOIN

An optimized plan

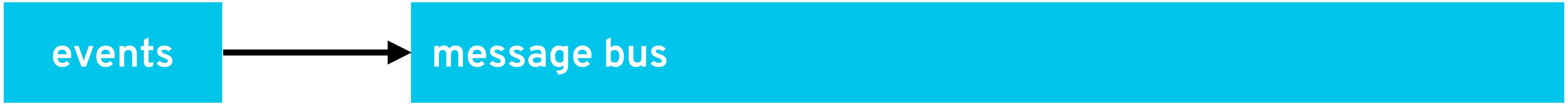


Structured streaming combines **stream processing** with **query planning** for **high-performance analytics on events**!

The Kappa architecture

events

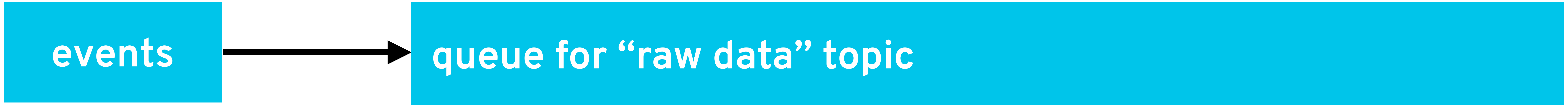
The Kappa architecture



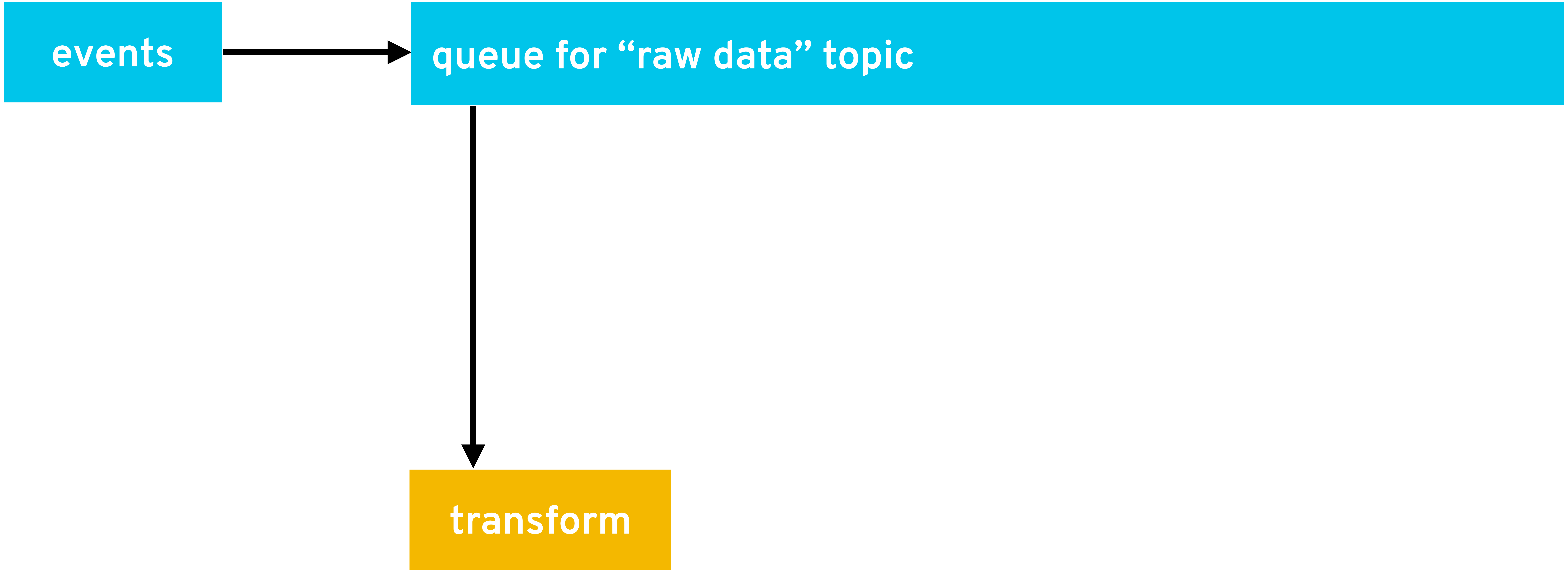
events

message bus

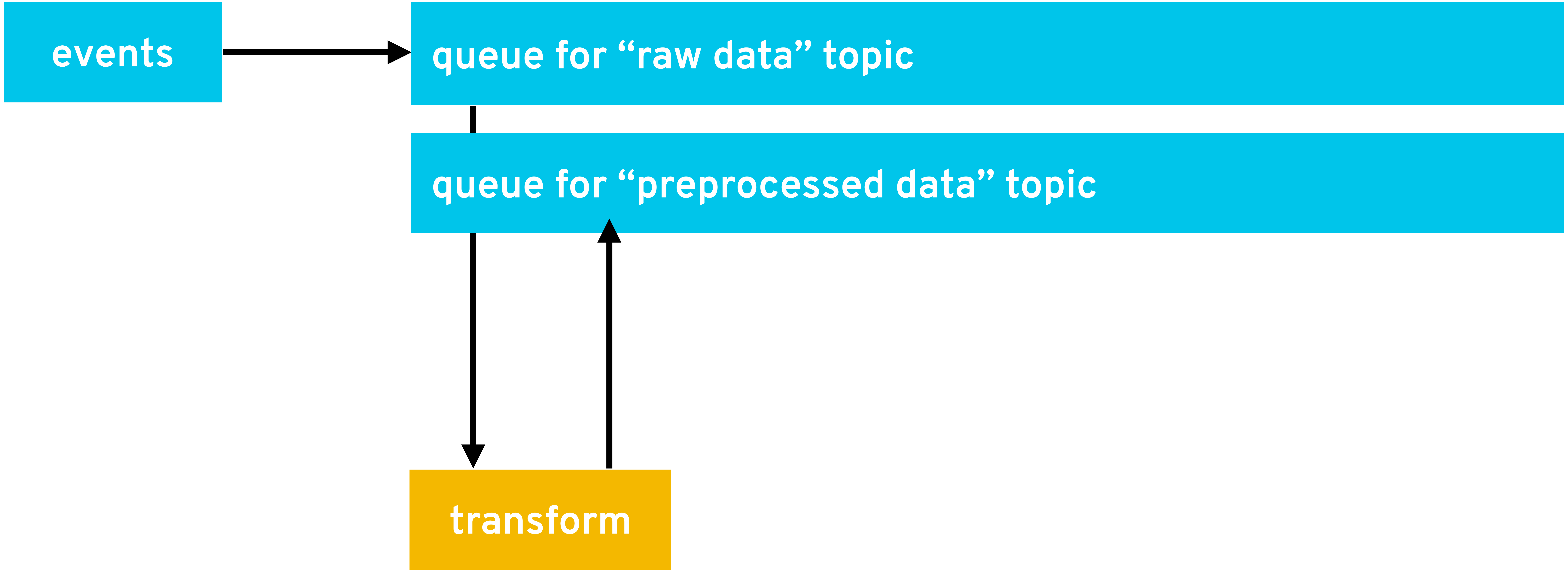
The Kappa architecture



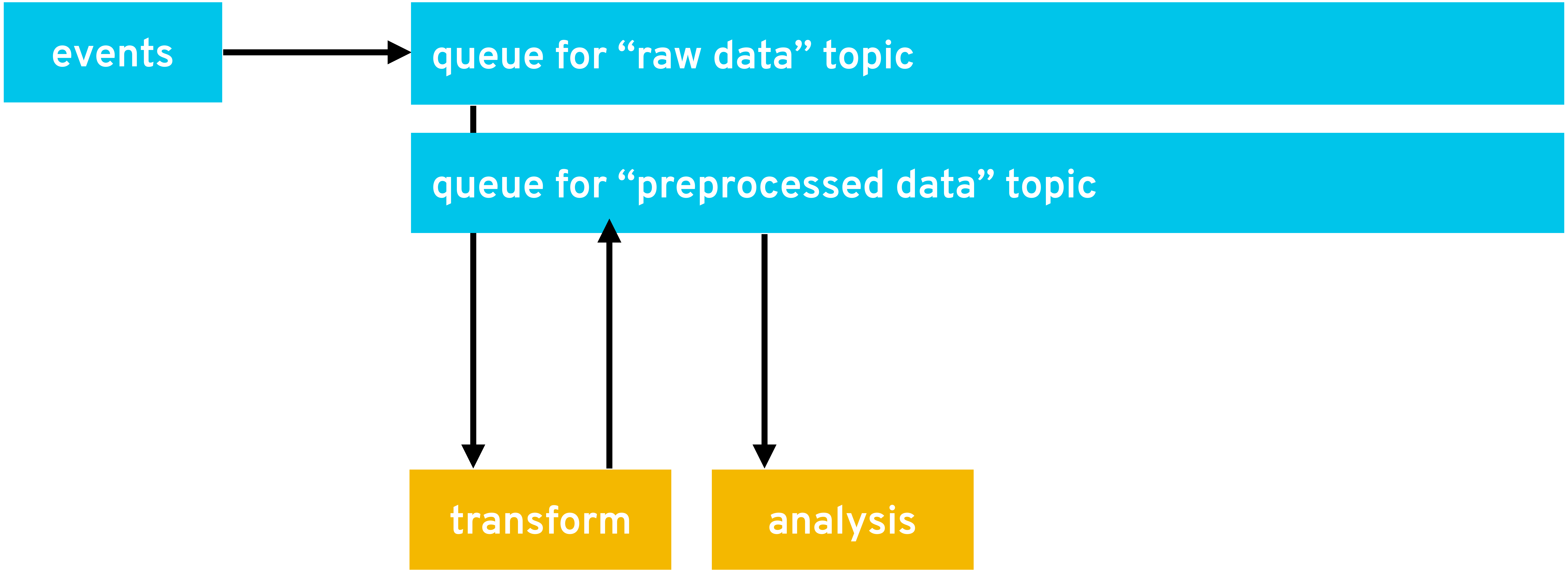
The Kappa architecture



The Kappa architecture


events
queue for “raw data” topic
queue for “preprocessed data” topic
transform

The Kappa architecture



events

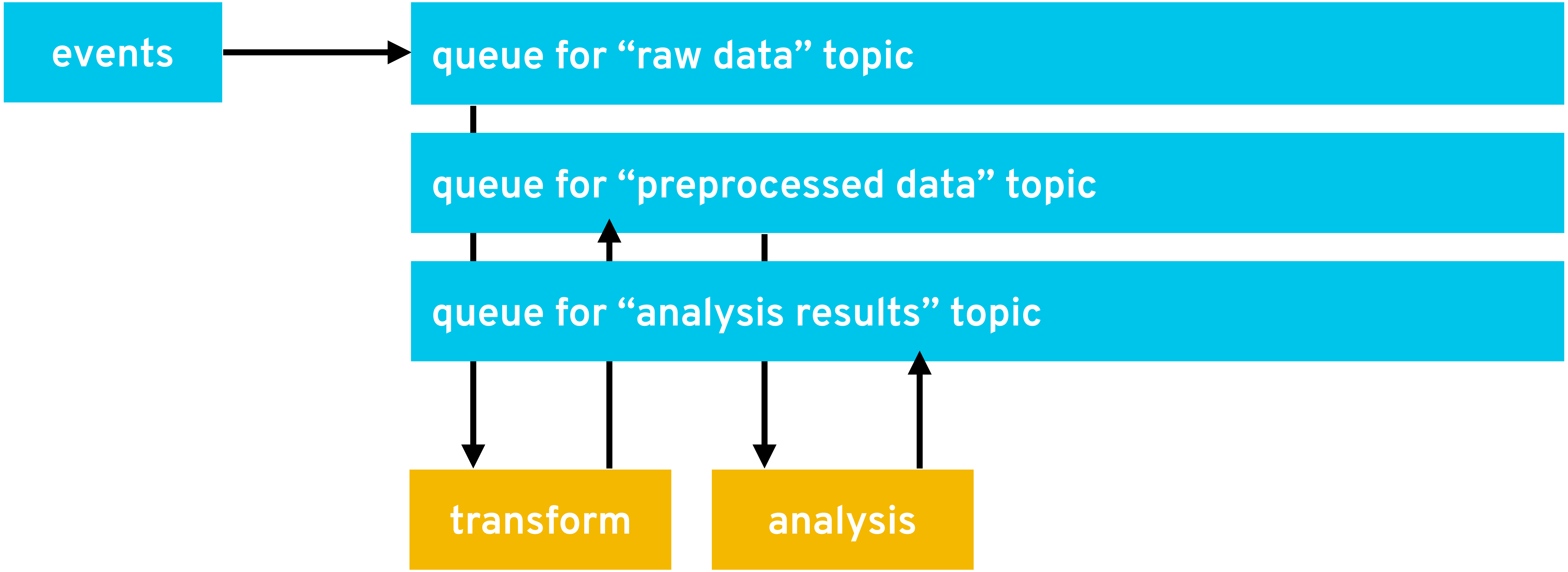
queue for “raw data” topic

queue for “preprocessed data” topic

transform

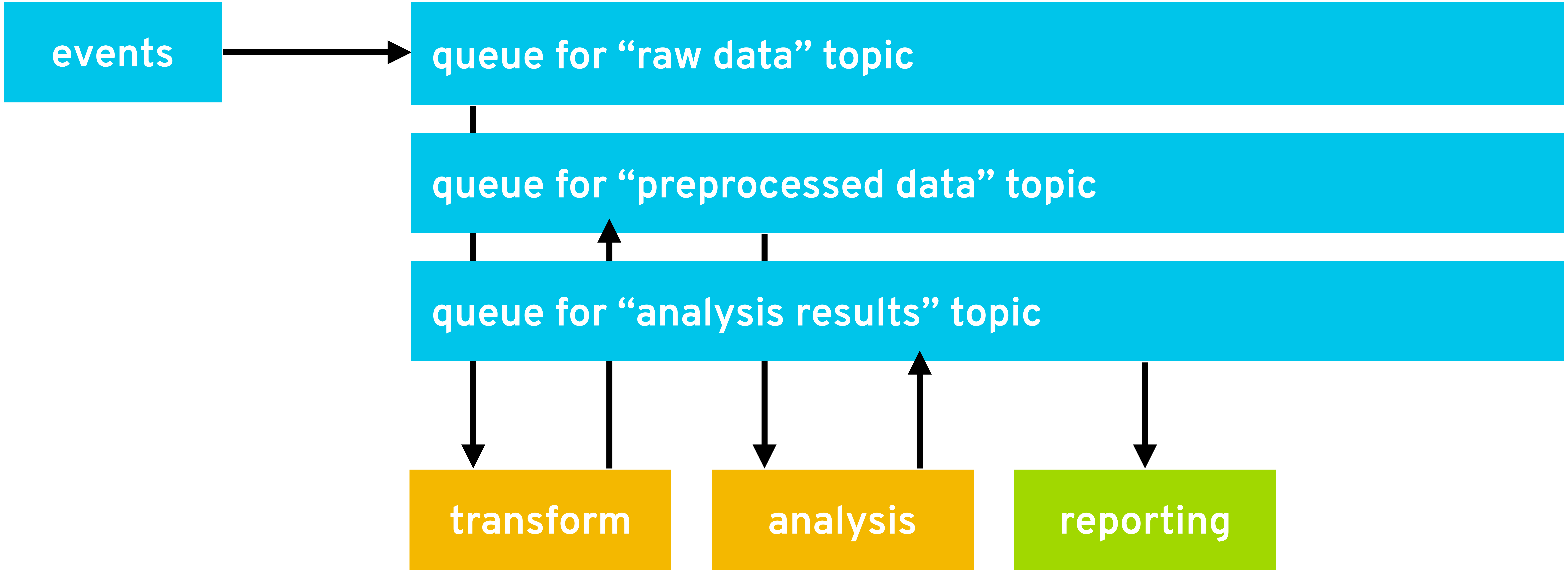
analysis

The Kappa architecture

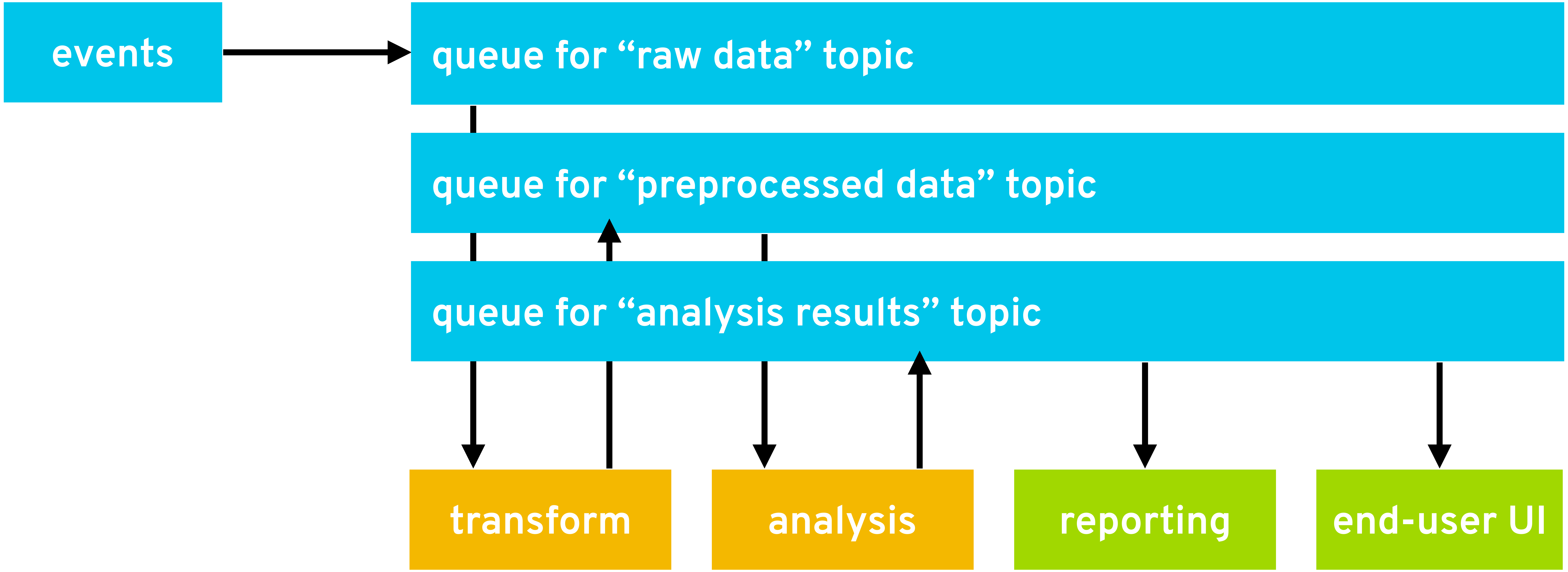


events
queue for “raw data” topic
queue for “preprocessed data” topic
queue for “analysis results” topic
transform
analysis

The Kappa architecture


events
queue for “raw data” topic
queue for “preprocessed data” topic
queue for “analysis results” topic
transform
analysis
reporting

The Kappa architecture



events

queue for “raw data” topic

queue for “preprocessed data” topic

queue for “analysis results” topic

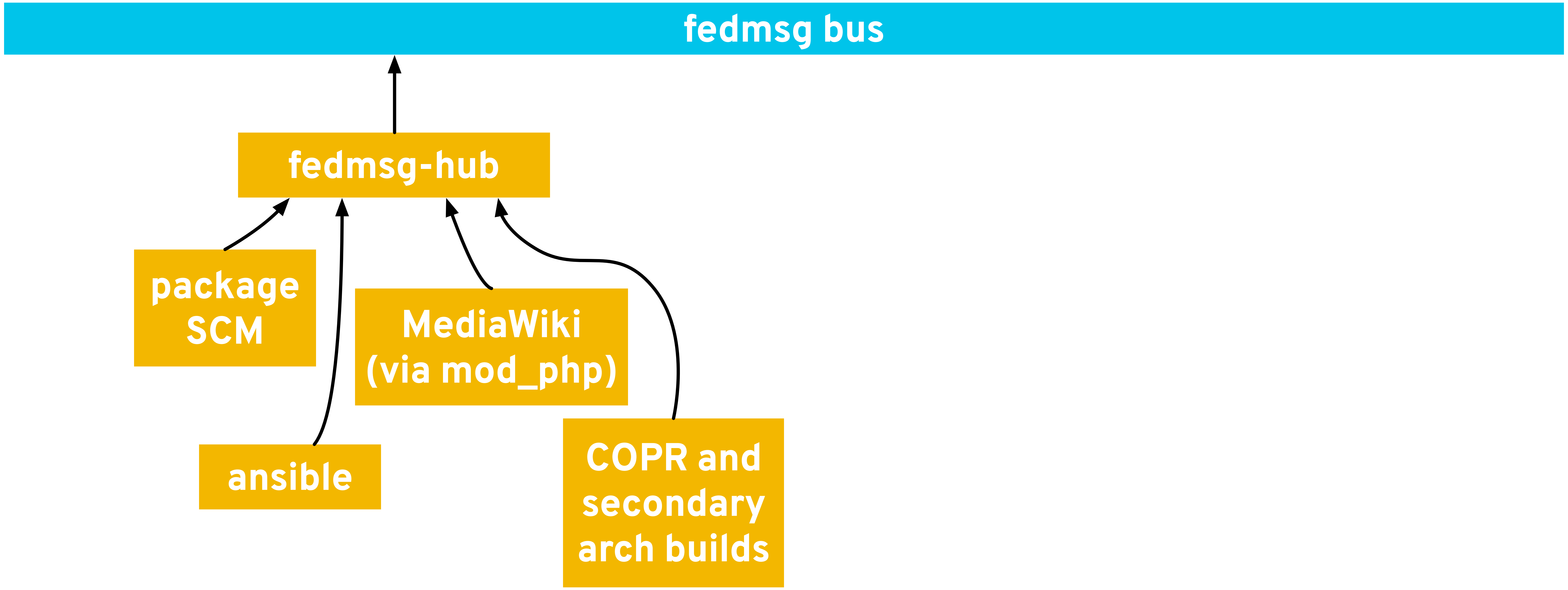
transform

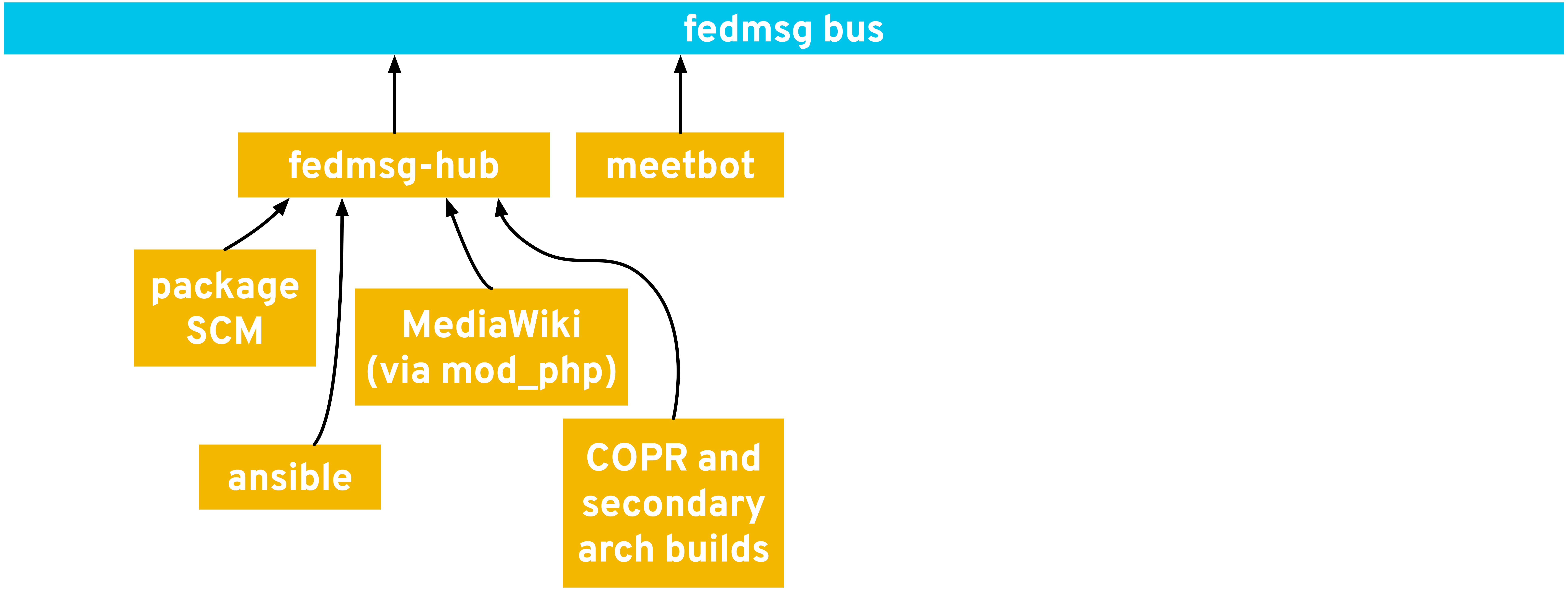
analysis

reporting

end-user UI

fedmsg bus


fedmsg bus
fedmsg-hub
package SCM
ansible
MediaWiki (via mod_php)
COPR and secondary arch builds



fedmsg bus

fedmsg-hub

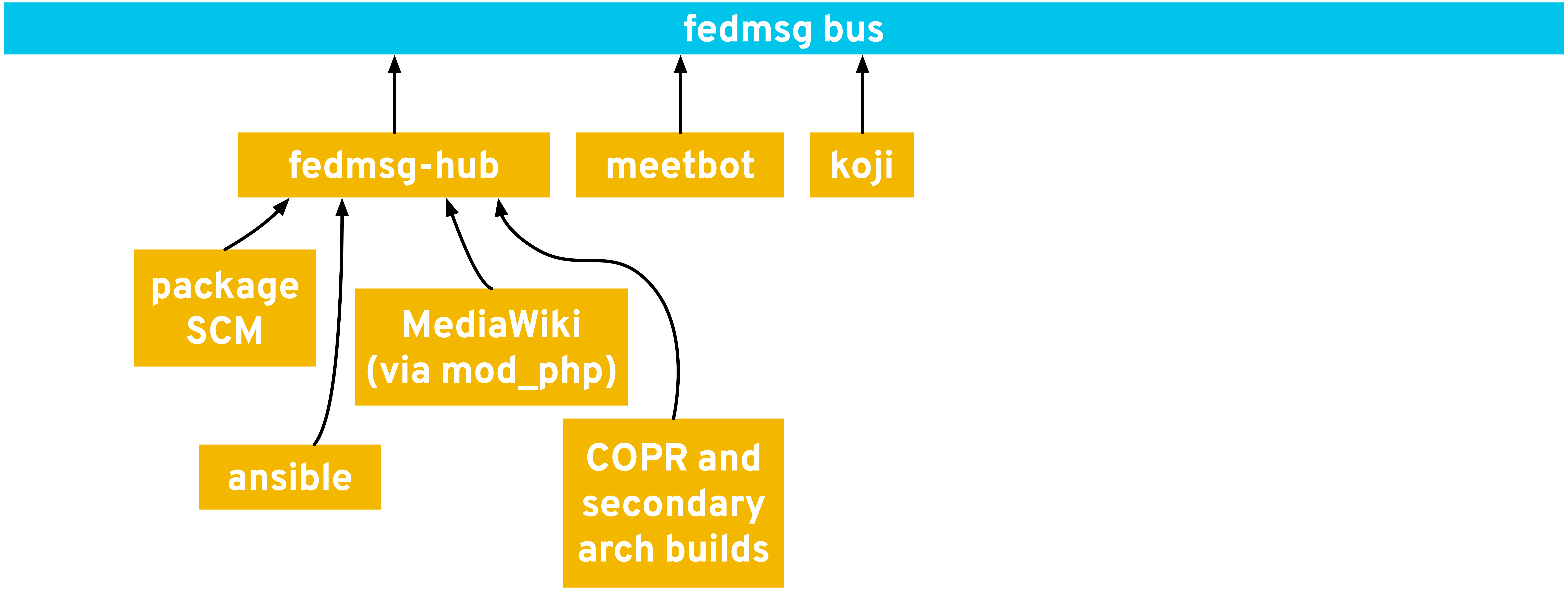
meetbot

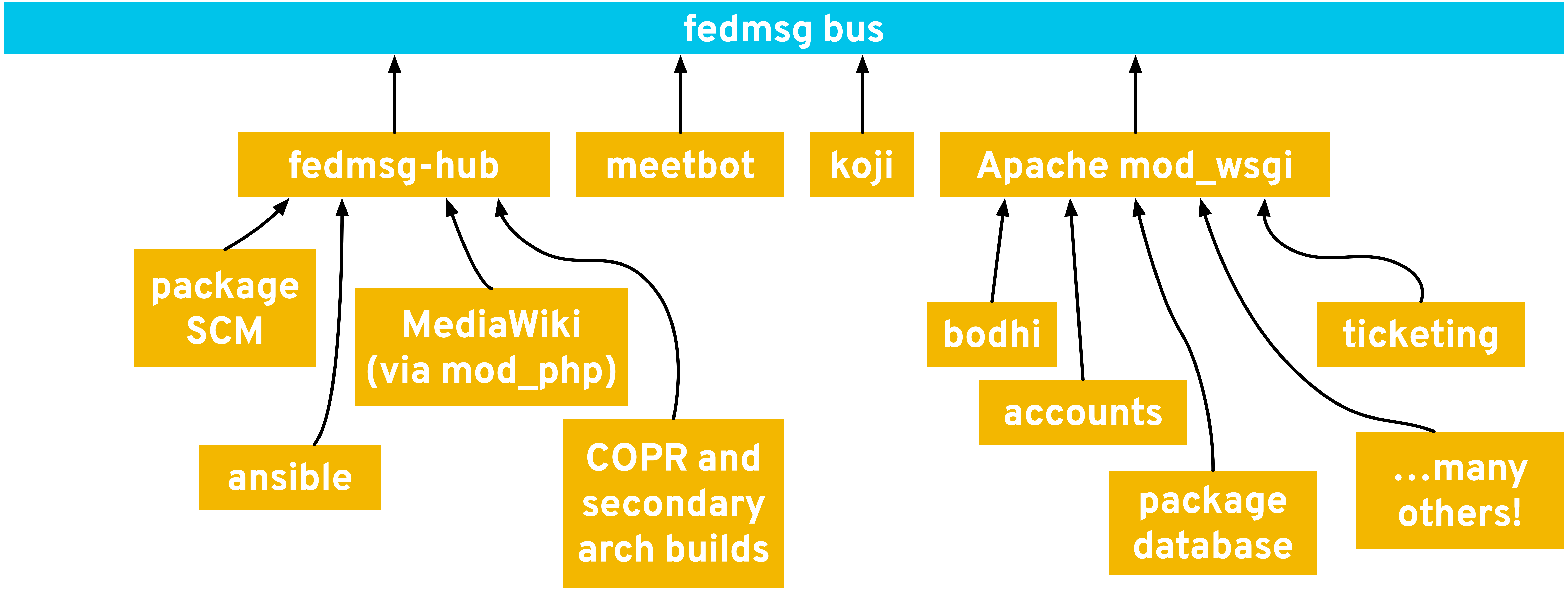
package SCM

MediaWiki (via mod_php)

ansible

COPR and secondary arch builds


fedmsg bus
fedmsg-hub
meetbot
koji
package SCM
MediaWiki (via mod_php)
ansible
COPR and secondary arch builds



fedmsg bus

fedmsg-hub

meetbot

koji

Apache mod_wsgi

package SCM

ansible

MediaWiki (via mod_php)

COPR and secondary arch builds

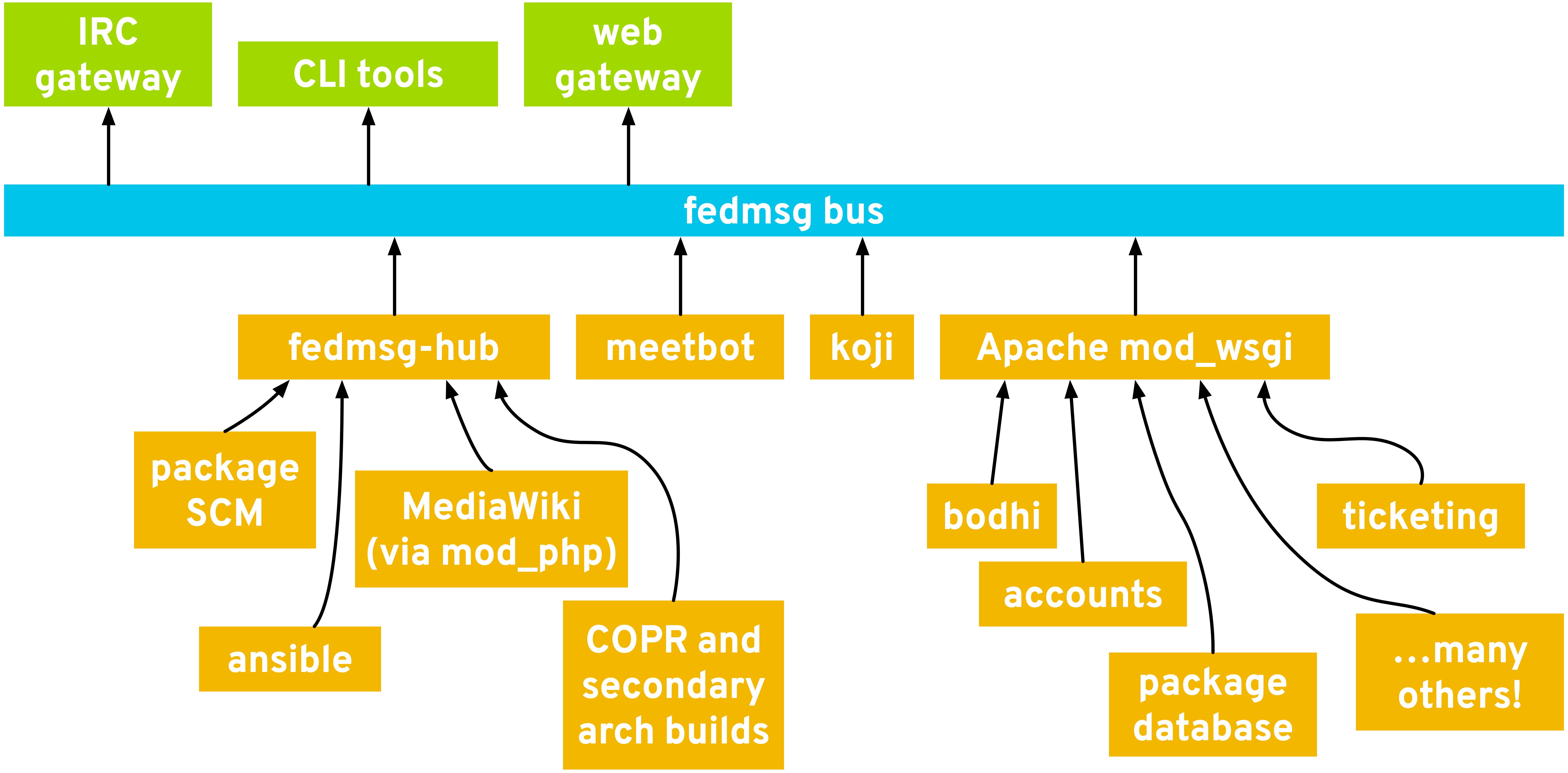
bodhi

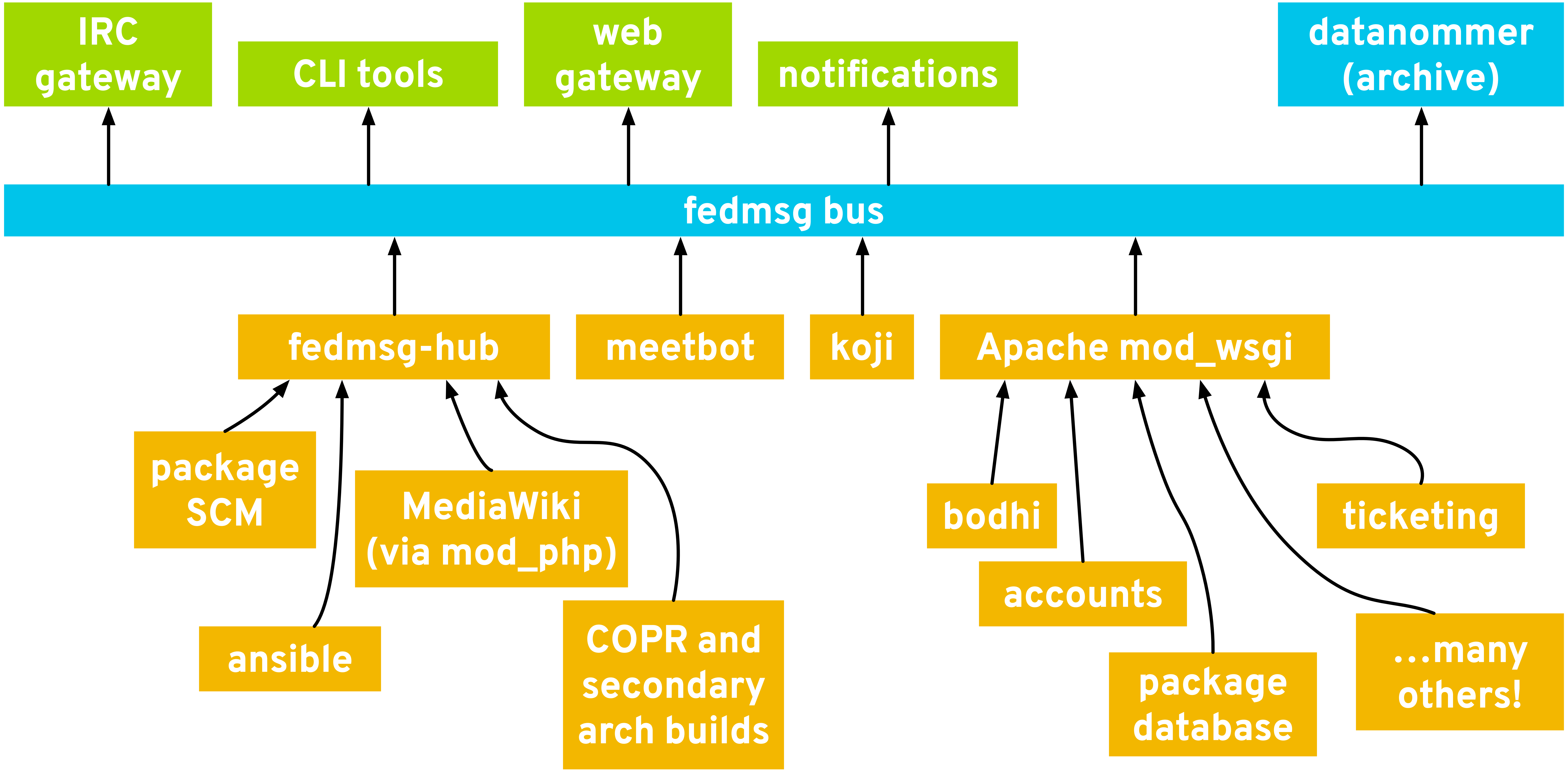
accounts

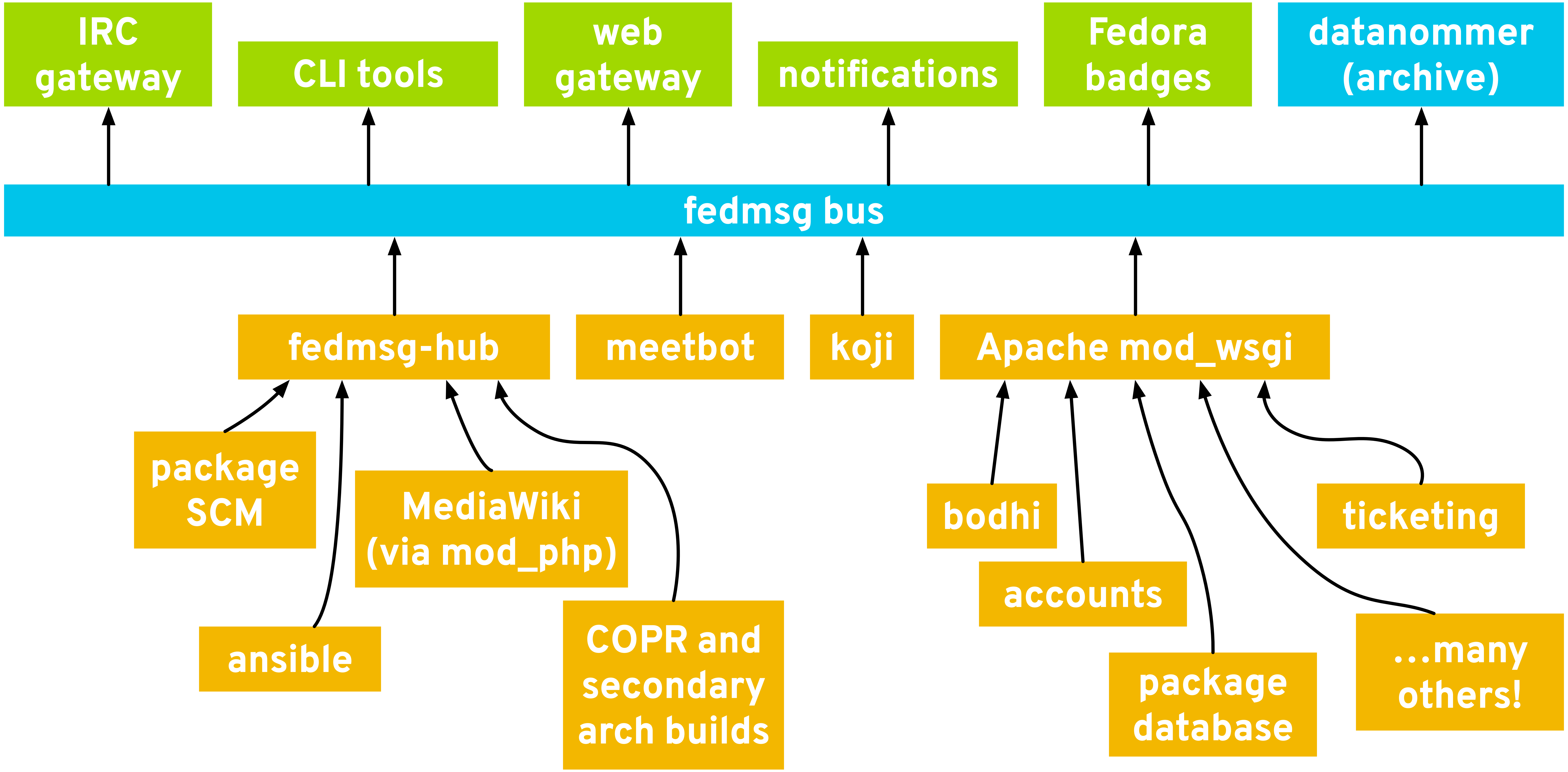
package database

ticketing

...many others!


IRC gateway
CLI tools
web gateway
fedmsg bus
fedmsg-hub
meetbot
koji
Apache mod_wsgi
package SCM
ansible
MediaWiki (via mod_php)
COPR and secondary arch builds
bodhi
accounts
package database
ticketing
...many others!


IRC gateway
CLI tools
web gateway
notifications
datanommer (archive)
fedmsg bus
fedmsg-hub
meetbot
koji
Apache mod_wsgi
package SCM
ansible
MediaWiki (via mod_php)
COPR and secondary arch builds
bodhi
accounts
package database
...many others!
ticketing


IRC gateway

CLI tools

web gateway

notifications

Fedora badges

datanommer (archive)

fedmsg bus

fedmsg-hub

meetbot

koji

Apache mod_wsgi

package SCM

ansible

MediaWiki (via mod_php)

COPR and secondary arch builds

bodhi

accounts

package database

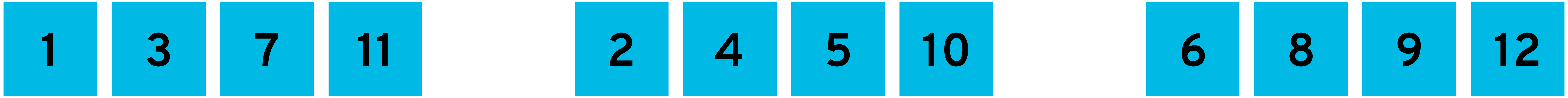
...many others!

ticketing

Apache Kafka

Apache Kafka

Apache Kafka

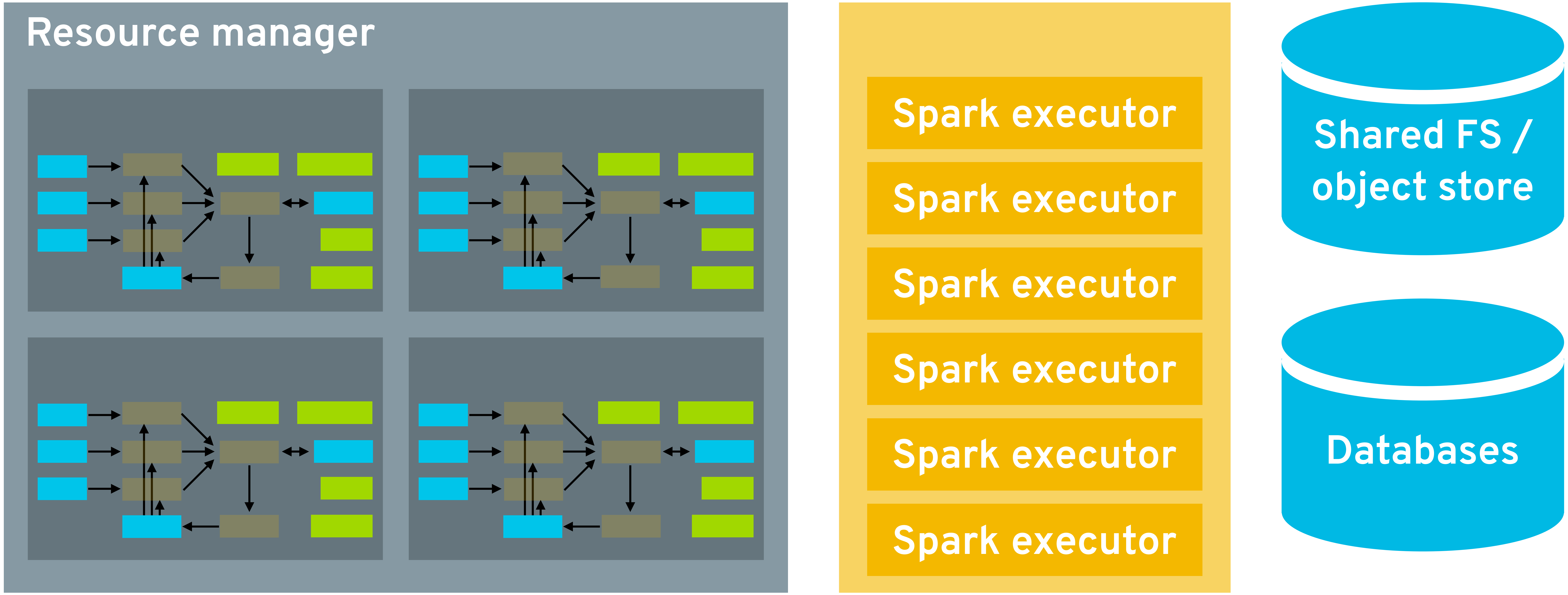


1 3 7 11

2 4 5 10

6 8 9 12

Multitenant compute clusters



Resource manager

Spark executor

Spark executor

Spark executor

Spark executor

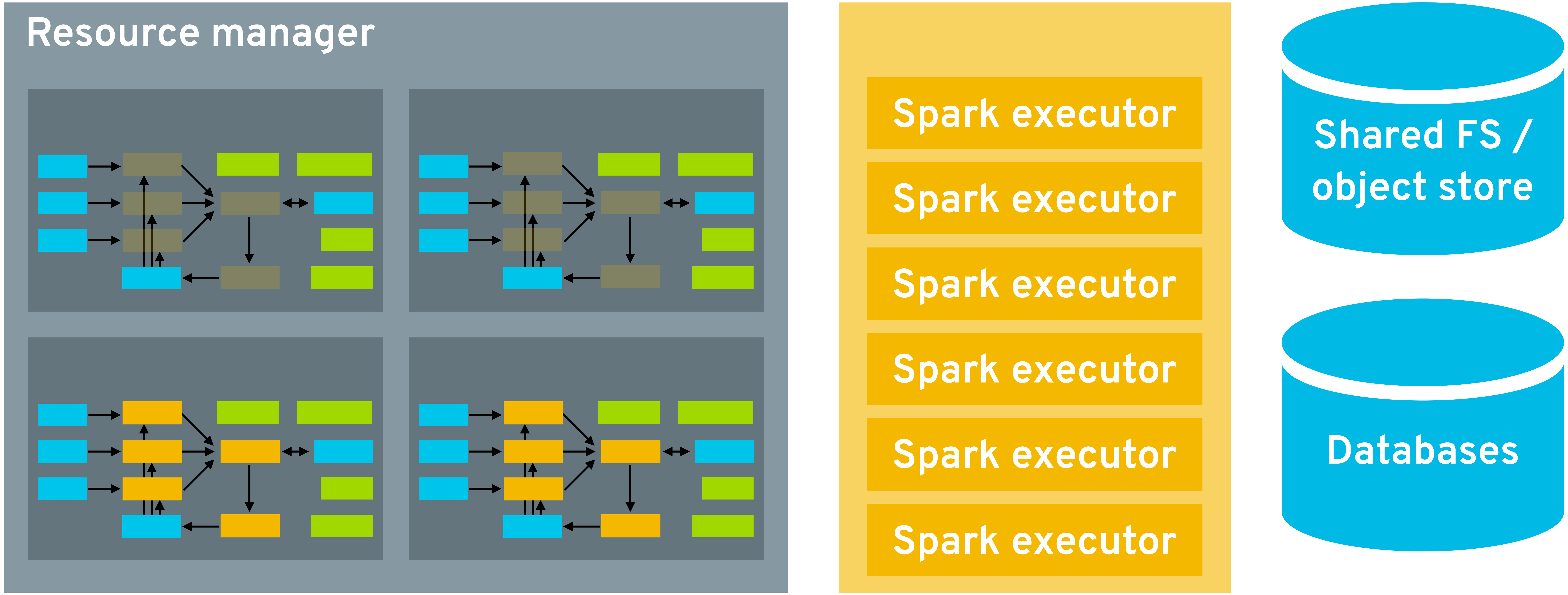
Spark executor

Spark executor

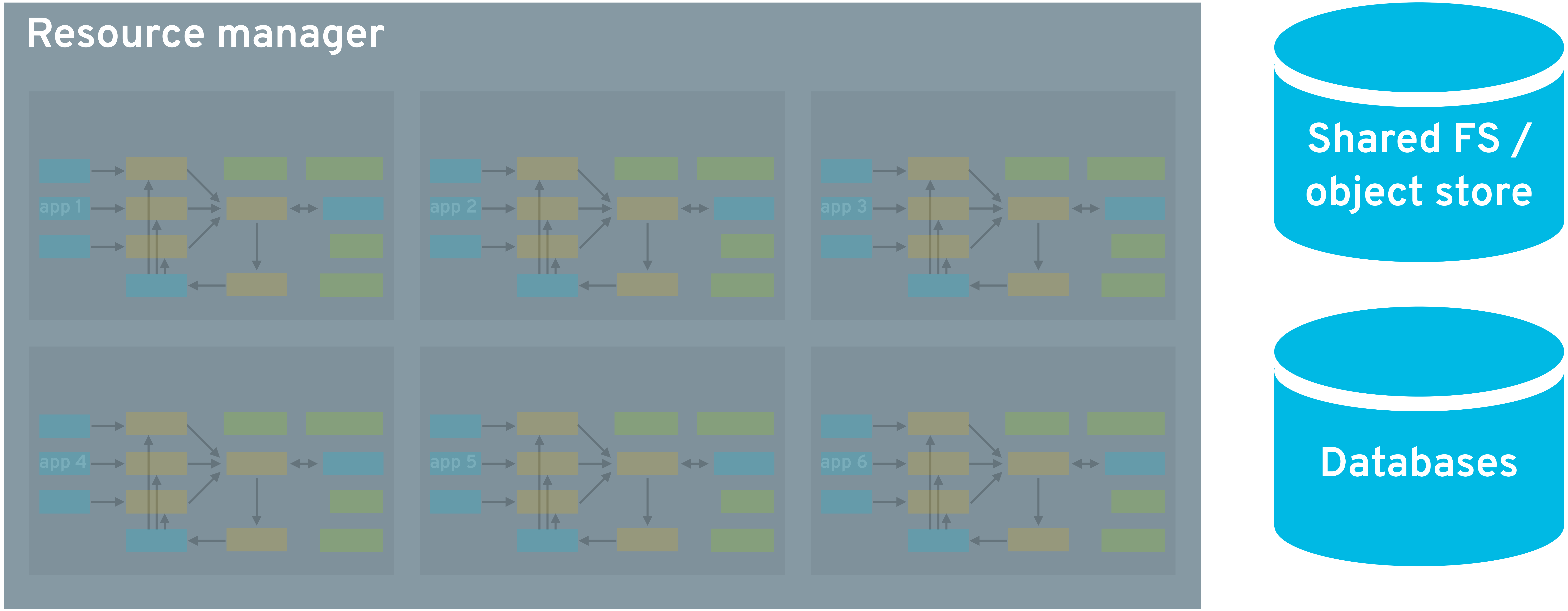
Shared FS / object store

Databases

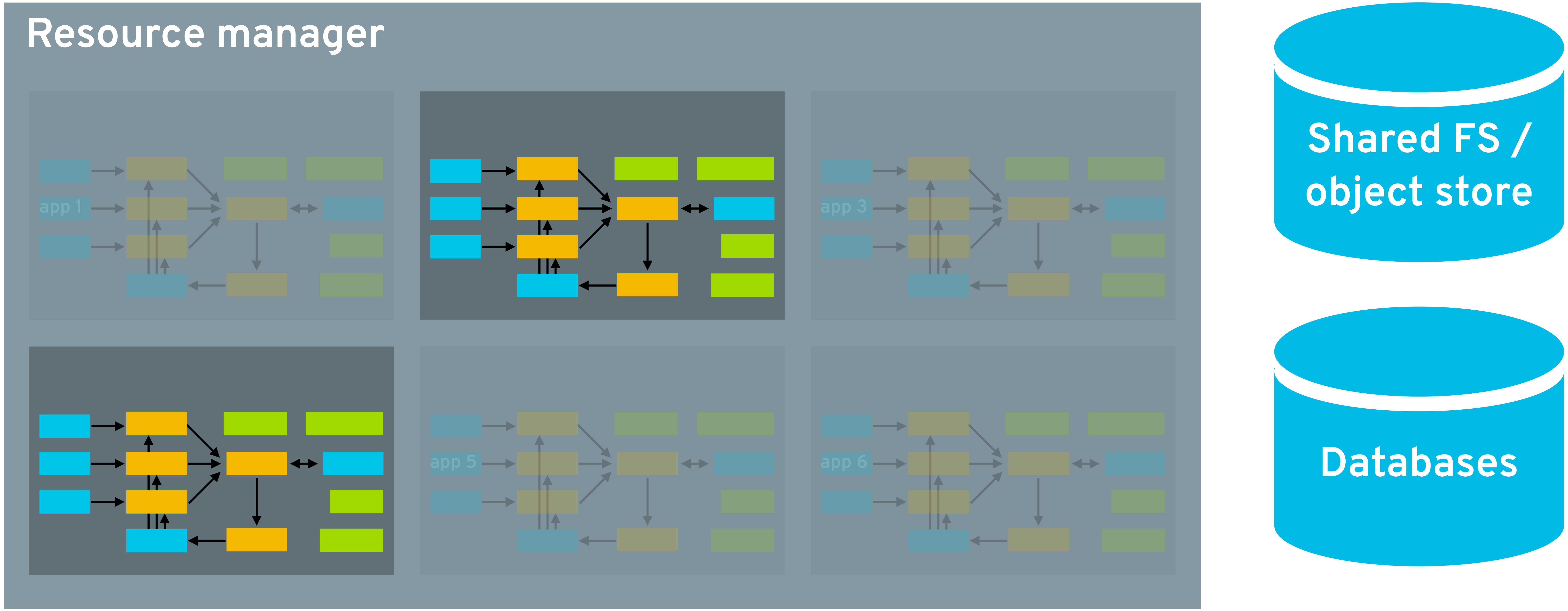
Multitenant compute clusters



One cluster per application



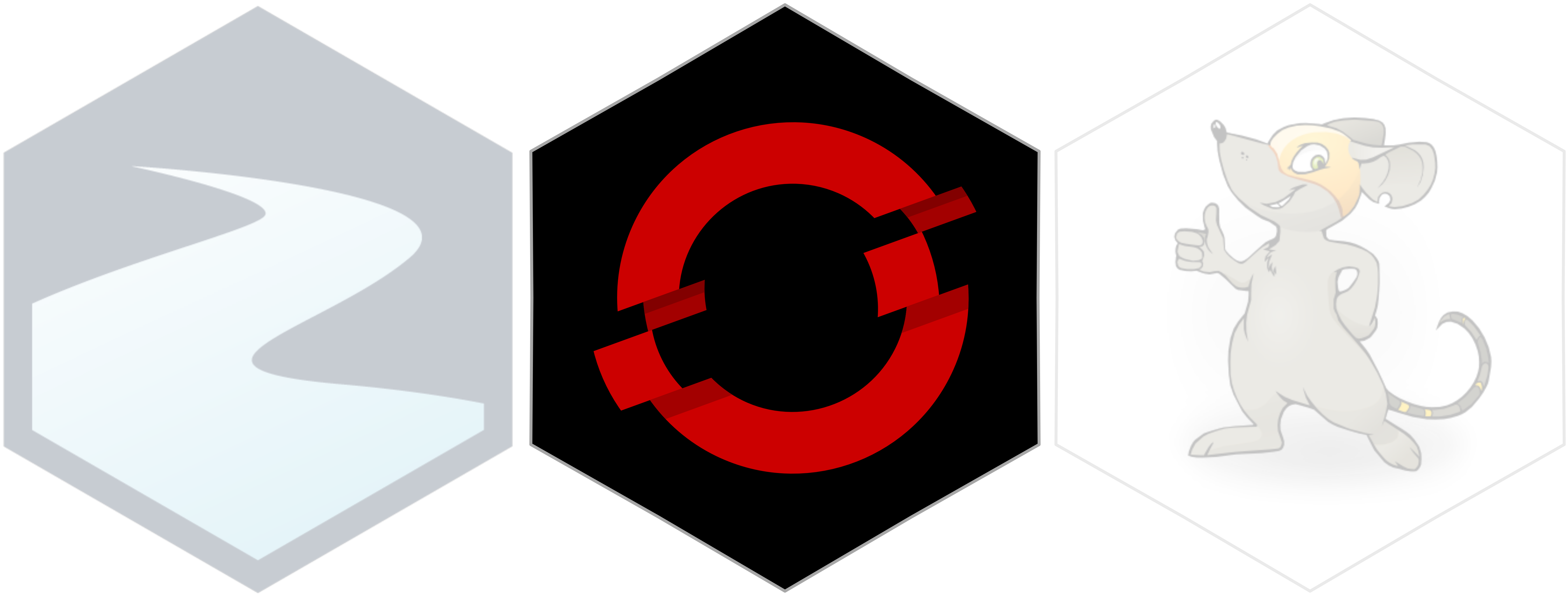
One cluster per application

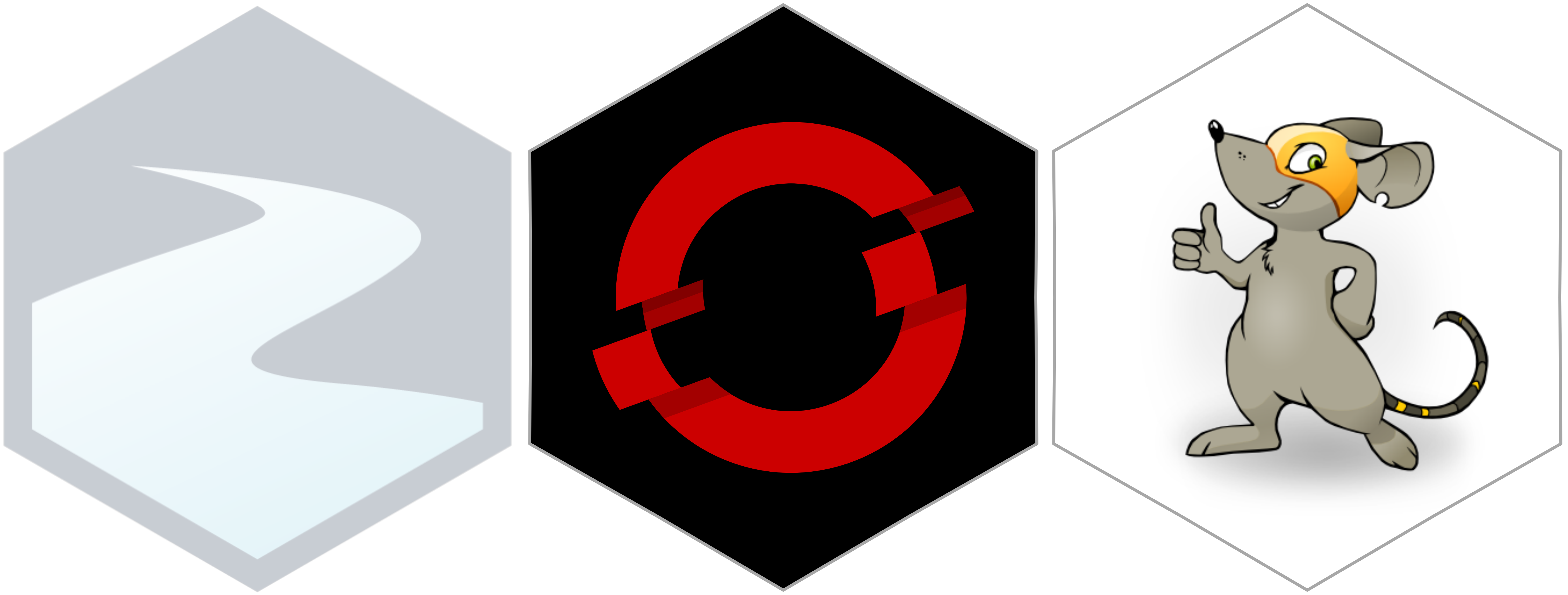


radanalytics.io

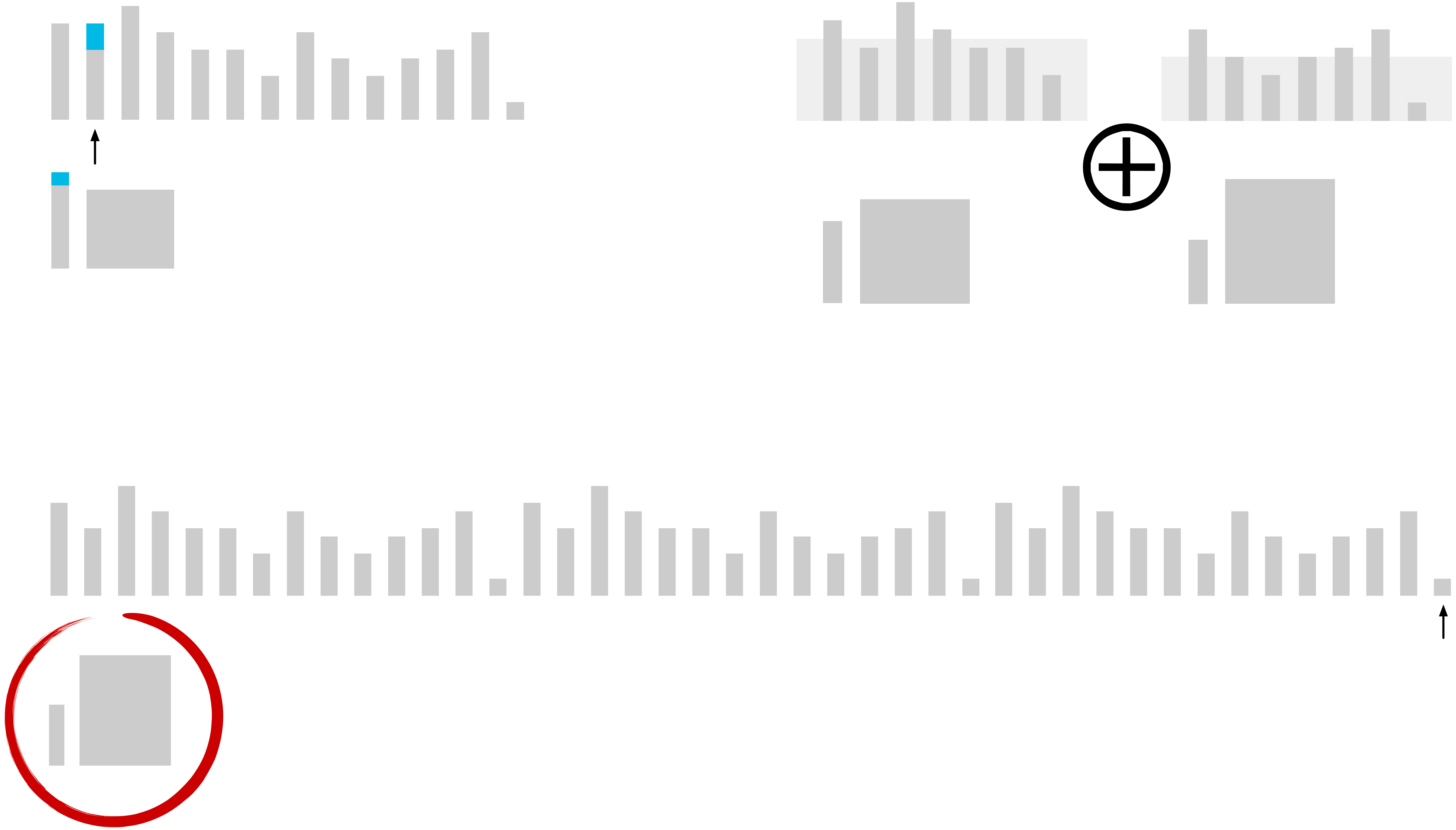
An **open-source community** enabling **intelligent applications** on **OpenShift**

Tooling to manage **Apache Spark**, **Jupyter notebooks**, and **TensorFlow training** and **model serving**





Properties of useful streaming algorithms



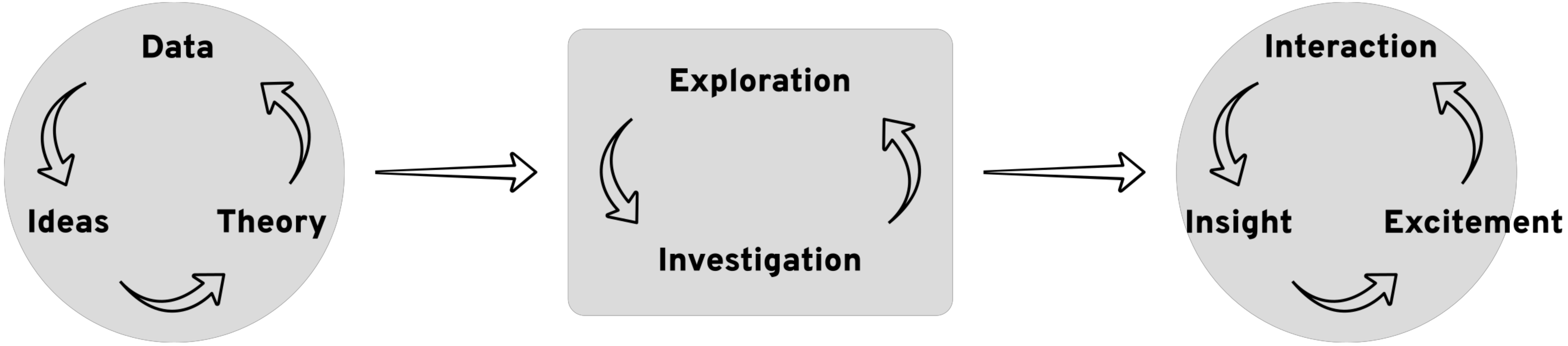
Interactivity time!

http://bit.ly/streaming-bds18

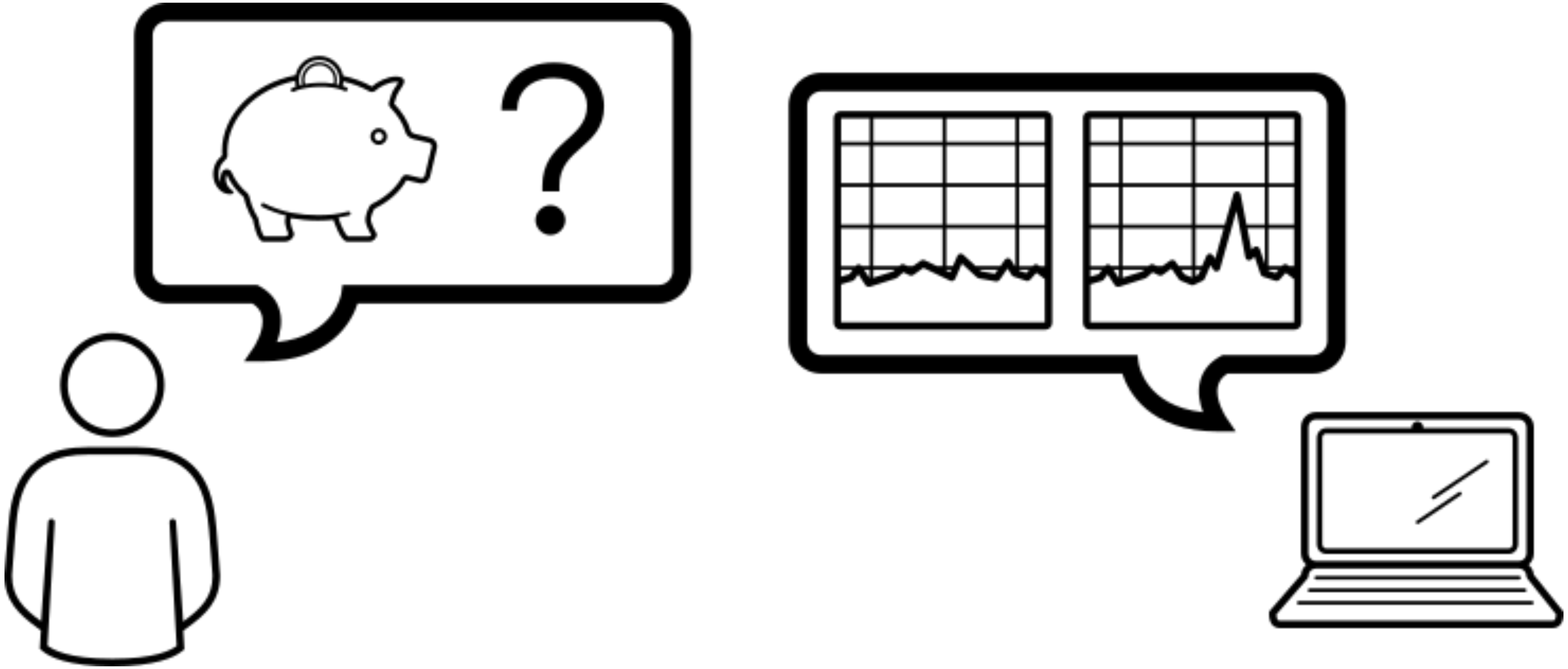
BIG DATA SPAIN

From notebooks and experiments to intelligent applications

Searching for solutions



Consider possible interactions



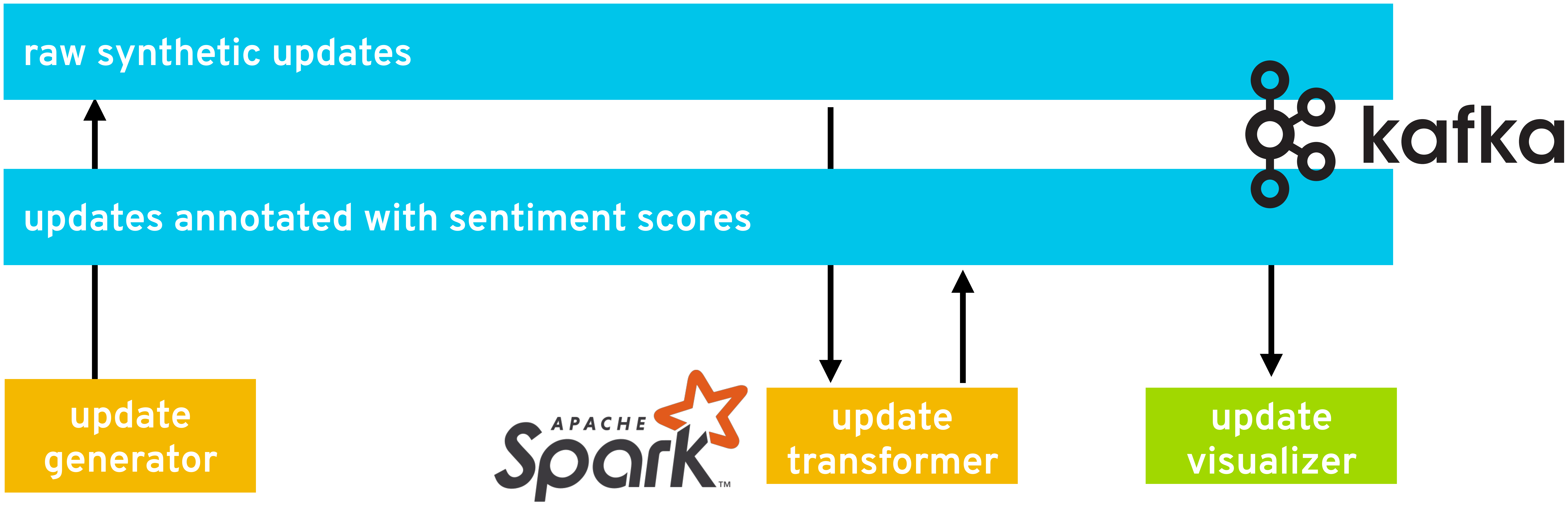
Plan for common patterns

Ingest

Process

Publish

Kappa in action



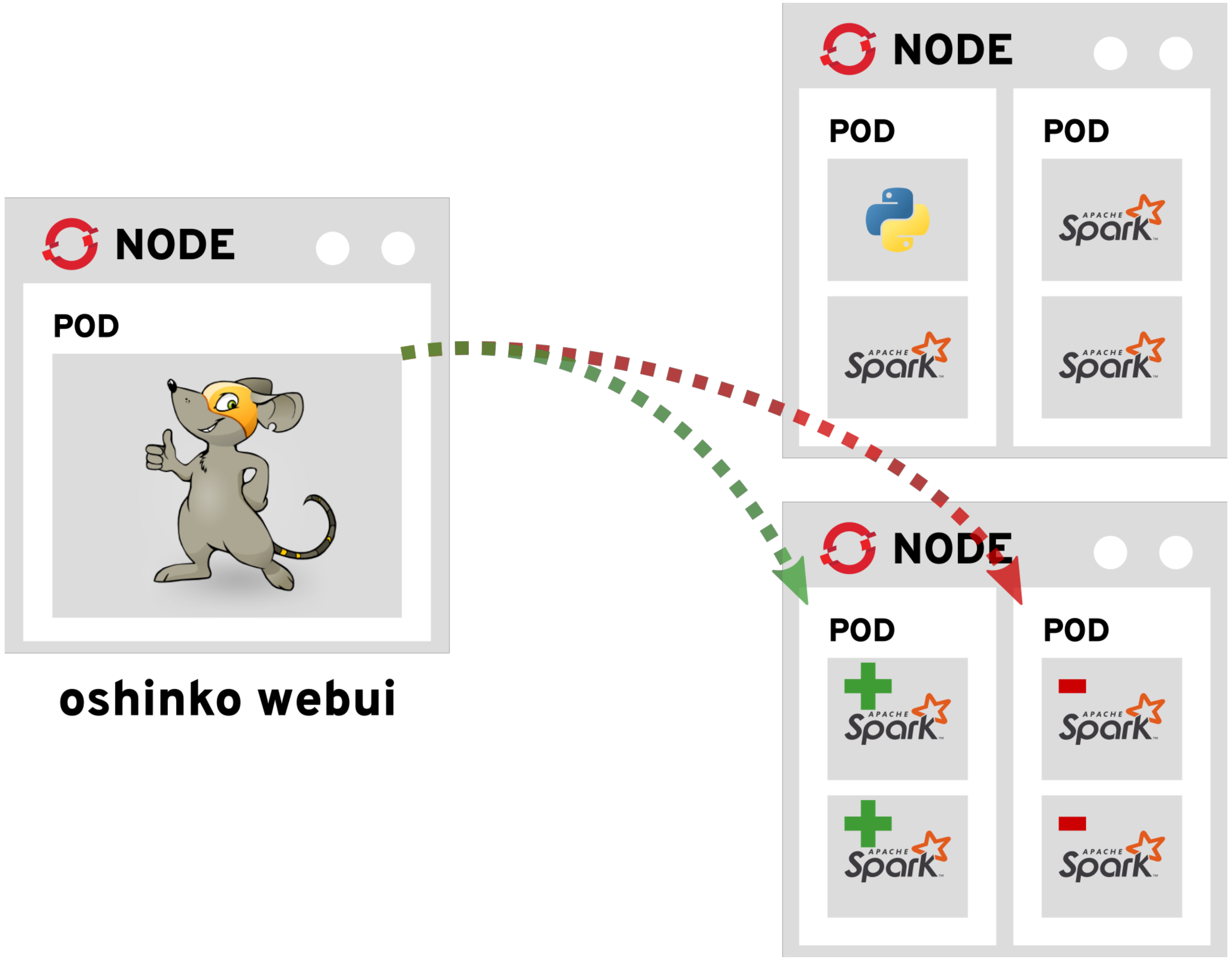
raw synthetic updates

updates annotated with sentiment scores

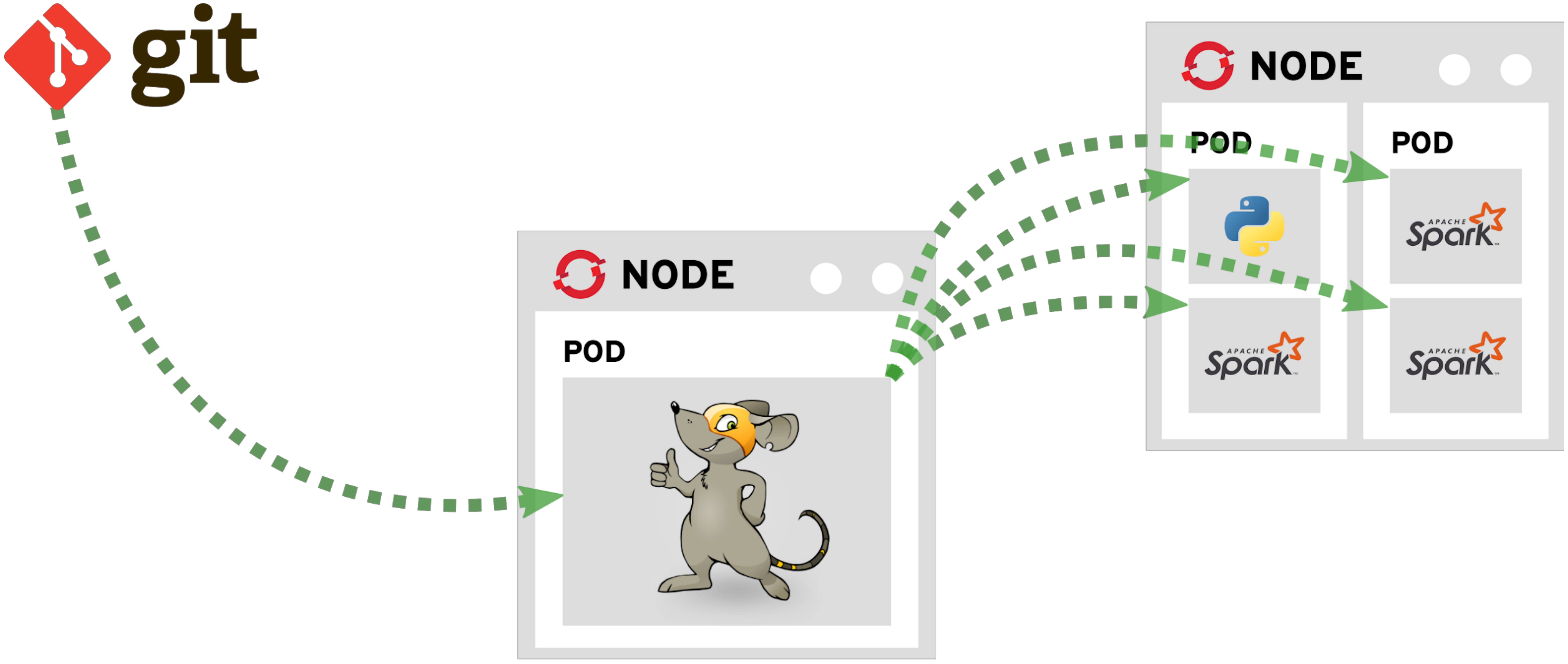
update generator

update transformer

update visualizer



oshinko webui



oshinko source-to-image