

# MARGARINE

---

## FEATURES

- ▶ URL to Shortened URL
- ▶ Inputting shortened URL redirects to URL
- ▶ Redis as data store

## POST: LOCALHOST:4000/

REQUEST

```
{"url": "http://elixiroutlaws.com/"}
```

RESPONSE 201

```
"margarine:hash:8fcc77a754"
```

RESPONSE 422

```
"Unable to shorten #{url}"
```

GET: LOCALHOST:4000/:KEY

```
GET localhost:4000/:key
```

https://elixiroutlaws.com

About Episodes Hosts Guests Contact Search



# Elixir Outlaws

The hallway track of the Elixir community

# ETS

---

## API SUBSET

- ▶ `insert(Tab, ObjectOrObjects) -> true`
- ▶ `insert_new(Tab, ObjectOrObjects) -> boolean()`
- ▶ `lookup(Tab, Key) -> [Object]`
- ▶ `new(Name, Options) -> tid\(\) | atom()`
- ▶ `whereis(TableName) -> tid() | undefined`

## EXAMPLE

```
iex(1)> :ets.new(:sample, [:named_table, :public, :set])
```

```
:sample
```

```
iex(2)> :ets.insert(:sample, {"key", "value"})
```

```
true
```

```
iex(3)> :ets.lookup(:sample, "key")
```

```
[{"key", "value"}]
```

# EXERCISE

---



# ADD LOCAL CACHE

---

# DISTRIBUTED FEATURES

---

## DISTRIBUTED FEATURES

- ▶ Data Replication and Consistency across nodes
- ▶ Local cache with fallback to primary store

# PG2

---

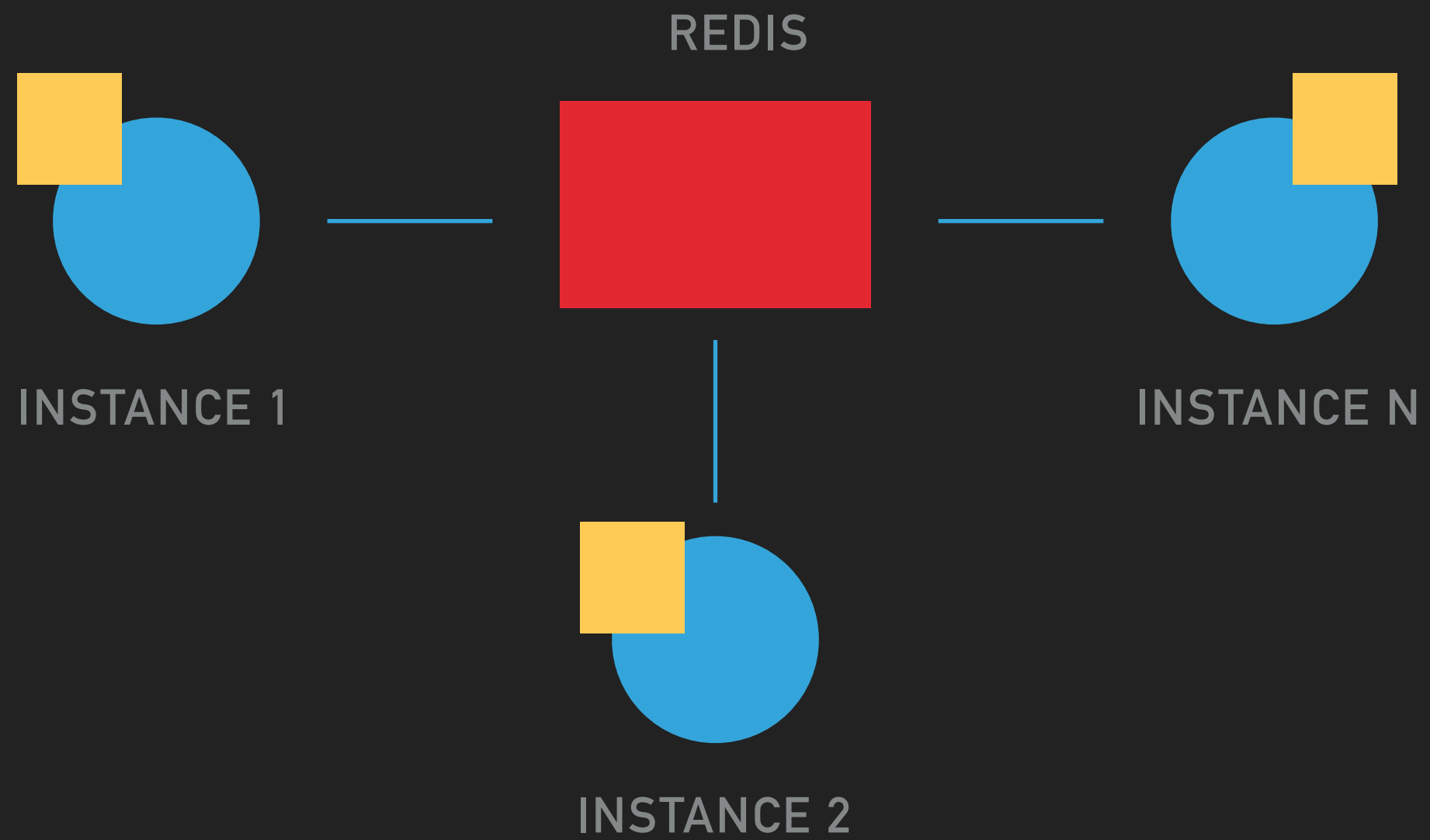
# API

- ▶ `create(Name :: name()) -> ok`
- ▶ `delete(Name :: name()) -> ok`
- ▶ `get_closest_pid(Name) -> pid() | {error, Reason}`
- ▶ `get_local_members(Name) -> [pid()] | {error, {no_such_group, Name}}`
- ▶ `get_members(Name) -> [pid()] | {error, {no_such_group, Name}}`
- ▶ `join(Name, Pid :: pid()) -> ok | {error, {no_such_group, Name}}`
- ▶ `leave(Name, Pid :: pid()) -> ok | {error, {no_such_group, Name}}`
- ▶ `start() / start_link() -> {ok, pid()} | {error, any()}`
- ▶ `which_groups() -> [Name :: name()]`

- ▶ Group processes across all nodes
- ▶ AP: available and partition tolerant.

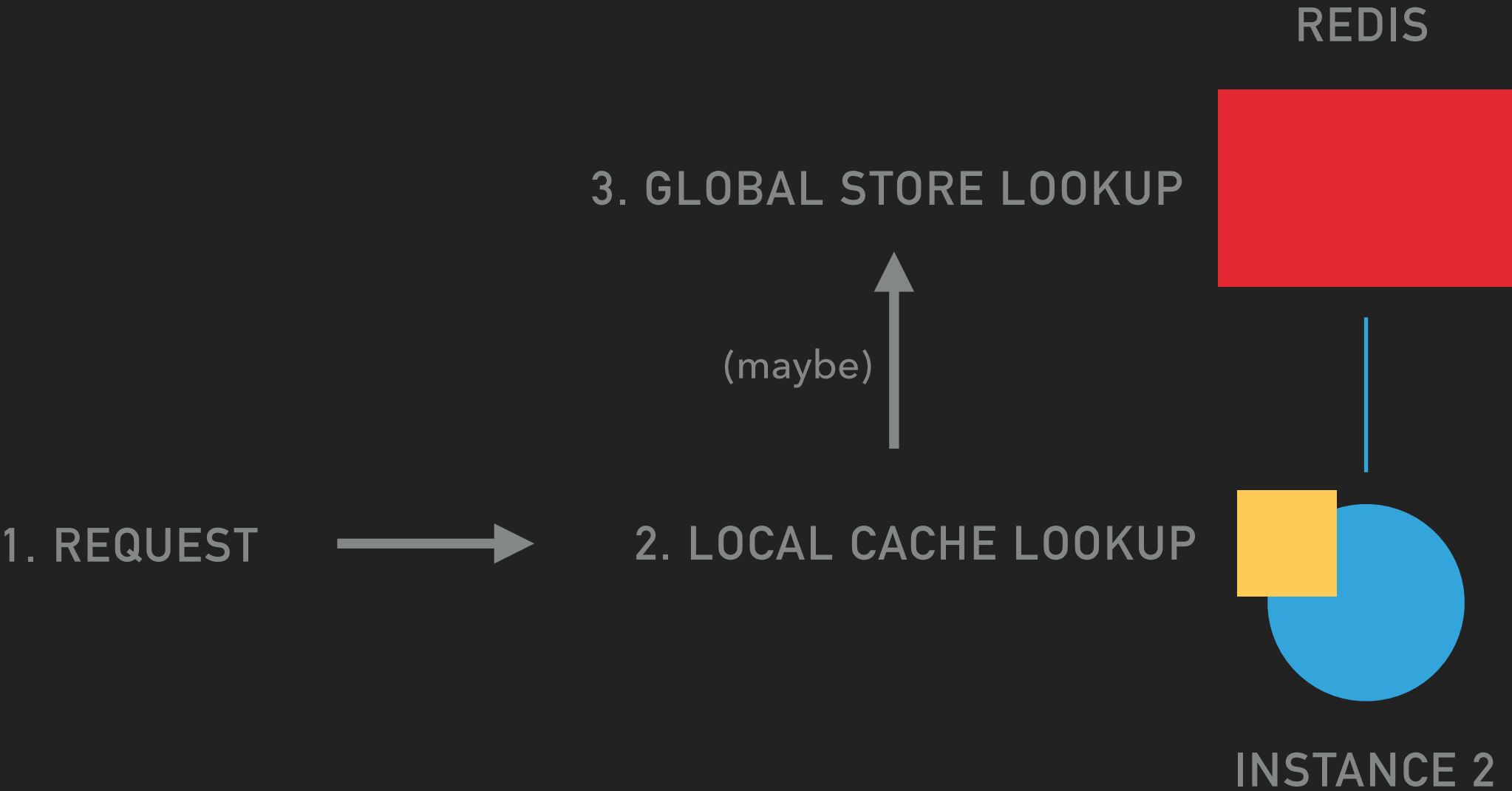
# DATA REPLICATION

---



# DATA REPLICATION

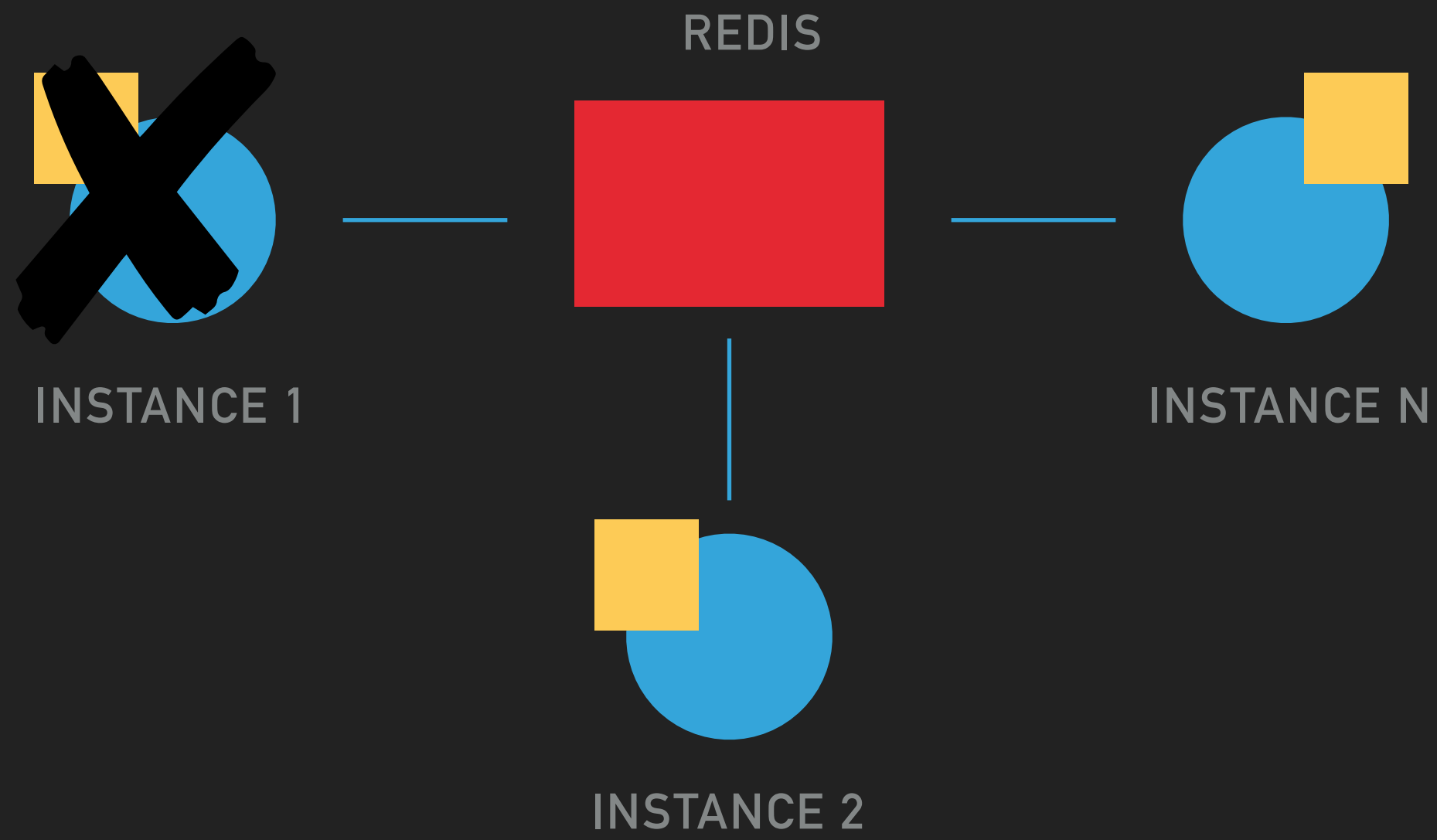
---





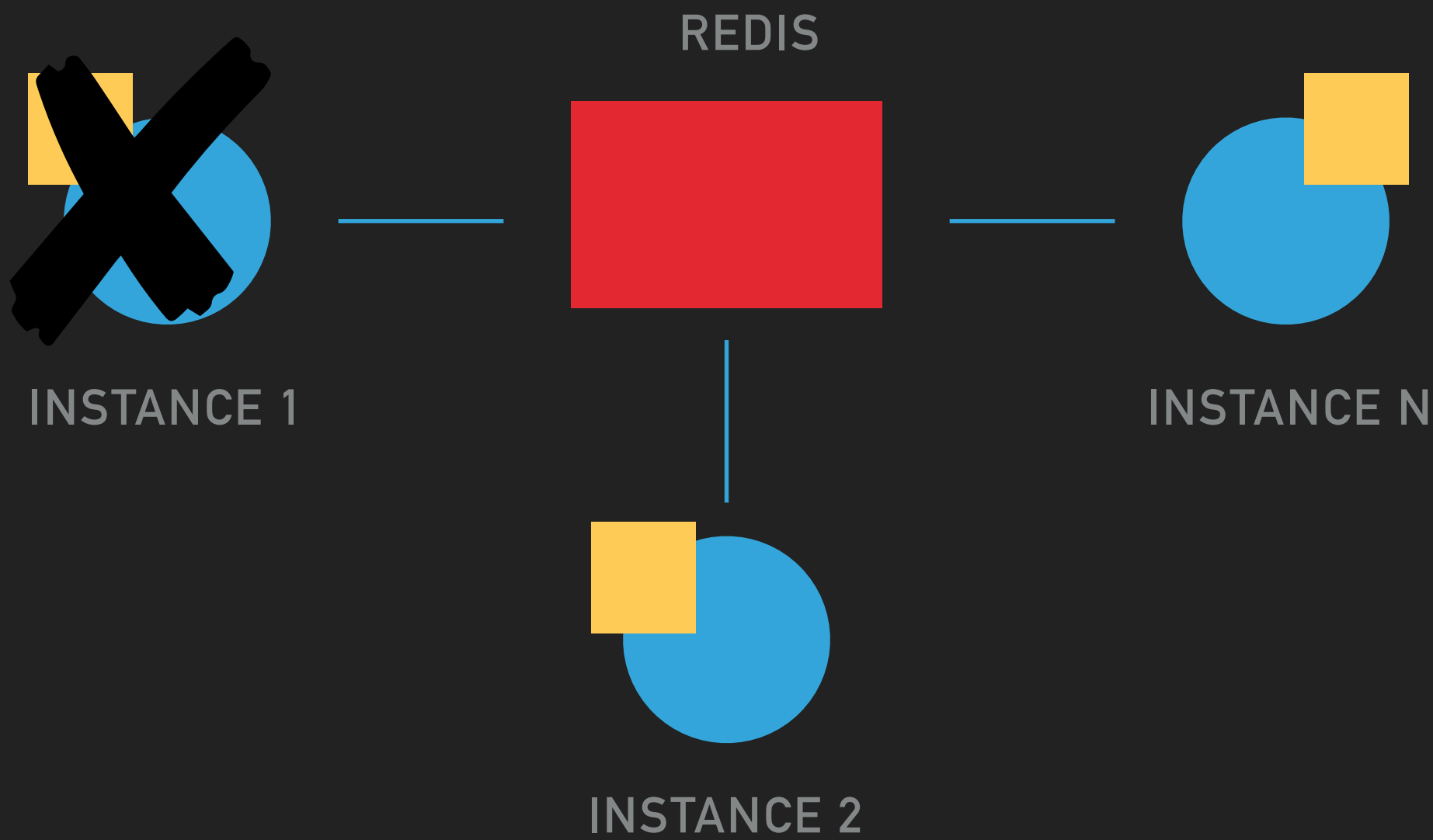
# NODE DISCONNECTION

---



# NODE CATCH UP

---



# EXERCISE

---