whatnot

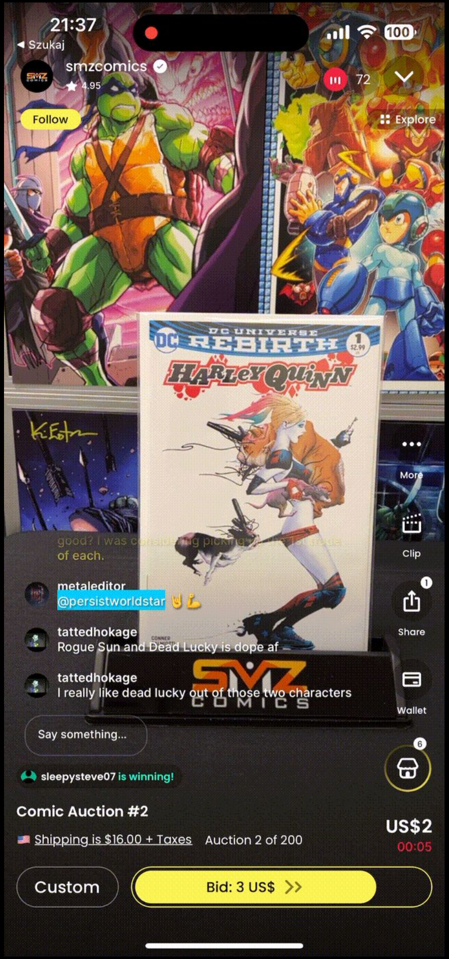
Keeping real-time auctions running during rollout

From white-knuckle to continuous deployments

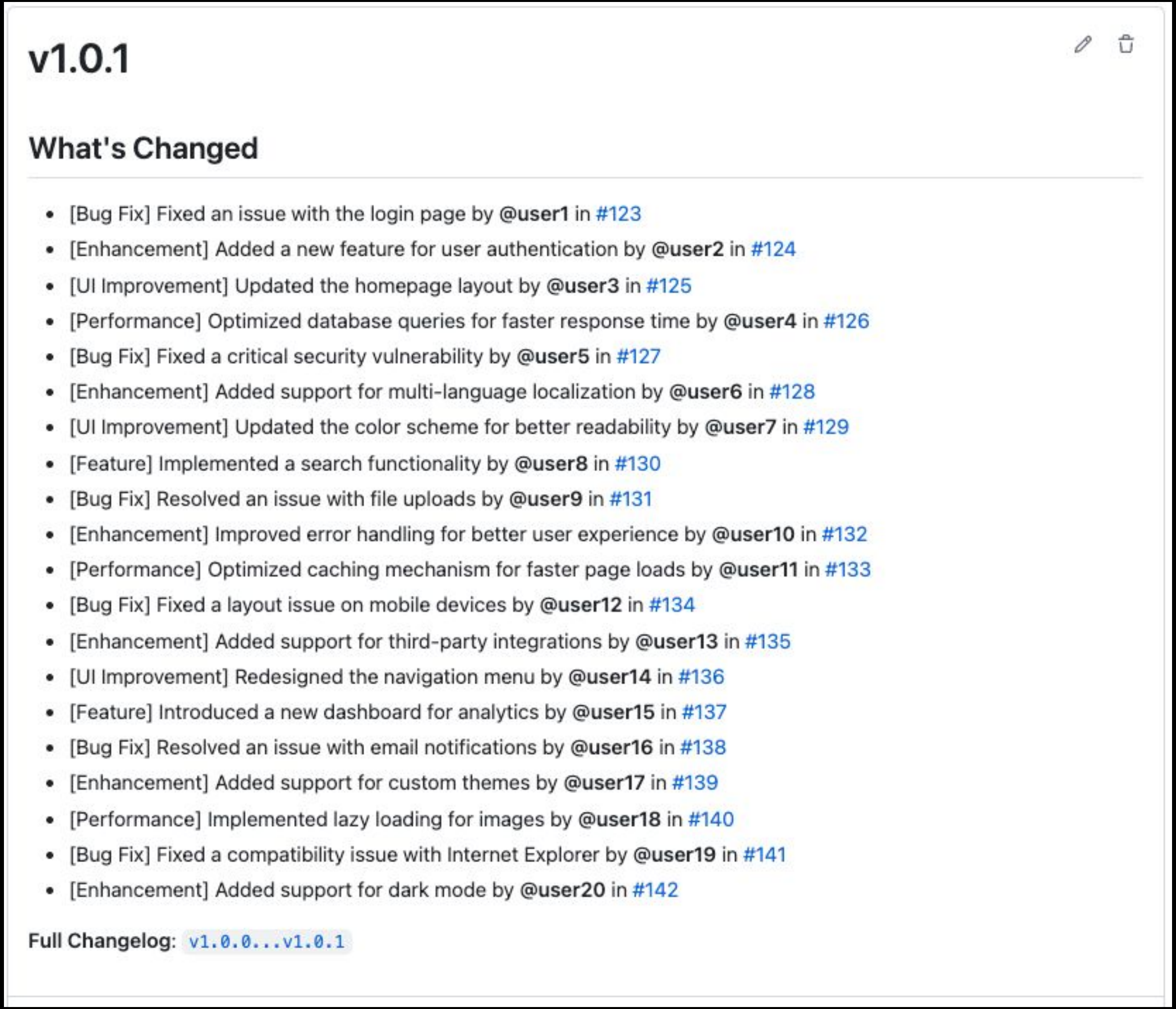
What is Whatnot?

What is Whatnot?

- A live shopping platform and marketplace to buy, sell and go live
- We're building a new kind of commerce experience that brings together community, shopping and entertainment
- Fastest-growing U.S. startup focused on livestream shopping



MARENVG

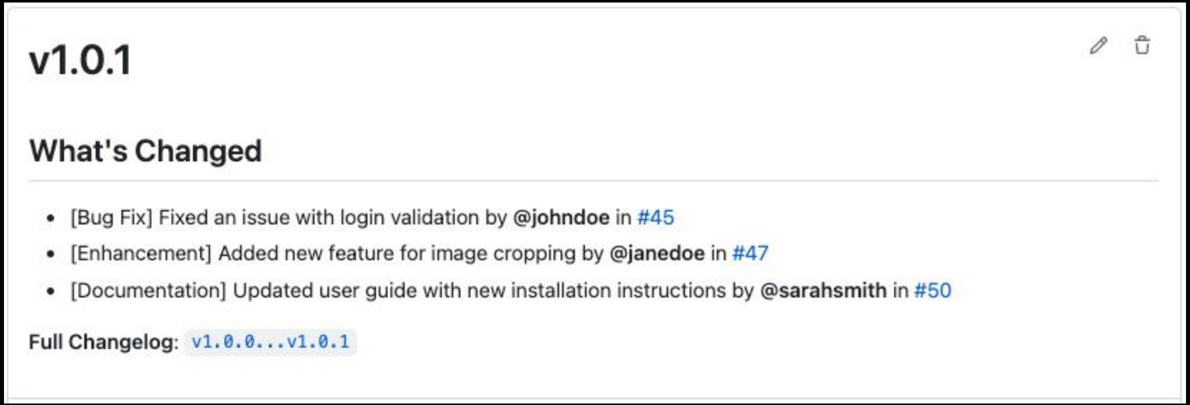


v1.0.1

What's Changed

- [Bug Fix] Fixed an issue with the login page by **@user1** in #123
- [Enhancement] Added a new feature for user authentication by **@user2** in #124
- [UI Improvement] Updated the homepage layout by **@user3** in #125
- [Performance] Optimized database queries for faster response time by **@user4** in #126
- [Bug Fix] Fixed a critical security vulnerability by **@user5** in #127
- [Enhancement] Added support for multi-language localization by **@user6** in #128
- [UI Improvement] Updated the color scheme for better readability by **@user7** in #129
- [Feature] Implemented a search functionality by **@user8** in #130
- [Bug Fix] Resolved an issue with file uploads by **@user9** in #131
- [Enhancement] Improved error handling for better user experience by **@user10** in #132
- [Performance] Optimized caching mechanism for faster page loads by **@user11** in #133
- [Bug Fix] Fixed a layout issue on mobile devices by **@user12** in #134
- [Enhancement] Added support for third-party integrations by **@user13** in #135
- [UI Improvement] Redesigned the navigation menu by **@user14** in #136
- [Feature] Introduced a new dashboard for analytics by **@user15** in #137
- [Bug Fix] Resolved an issue with email notifications by **@user16** in #138
- [Enhancement] Added support for custom themes by **@user17** in #139
- [Performance] Implemented lazy loading for images by **@user18** in #140
- [Bug Fix] Fixed a compatibility issue with Internet Explorer by **@user19** in #141
- [Enhancement] Added support for dark mode by **@user20** in #142

Full Changelog: `v1.0.0...v1.0.1`

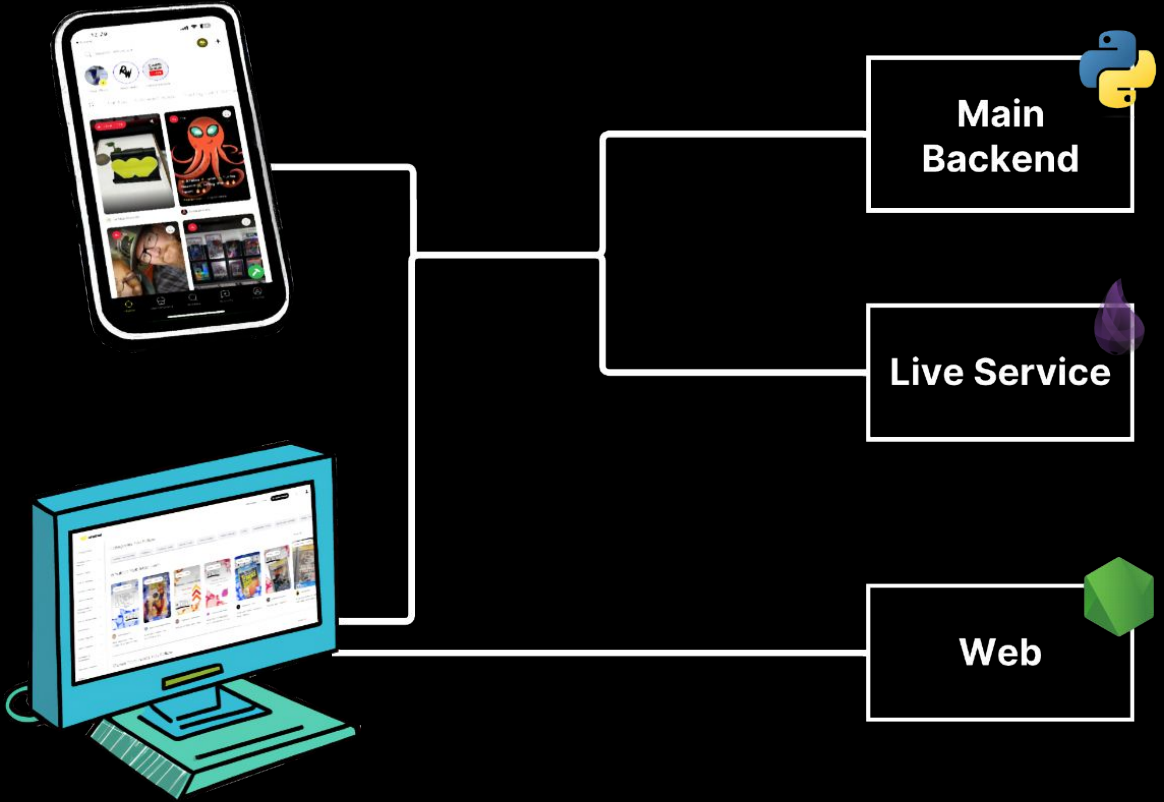


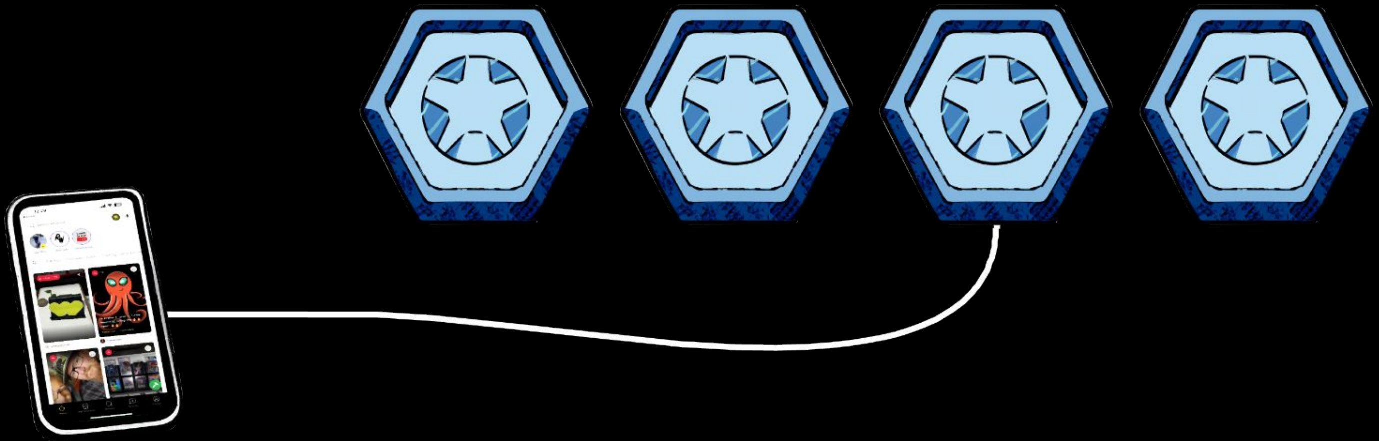
v1.0.1

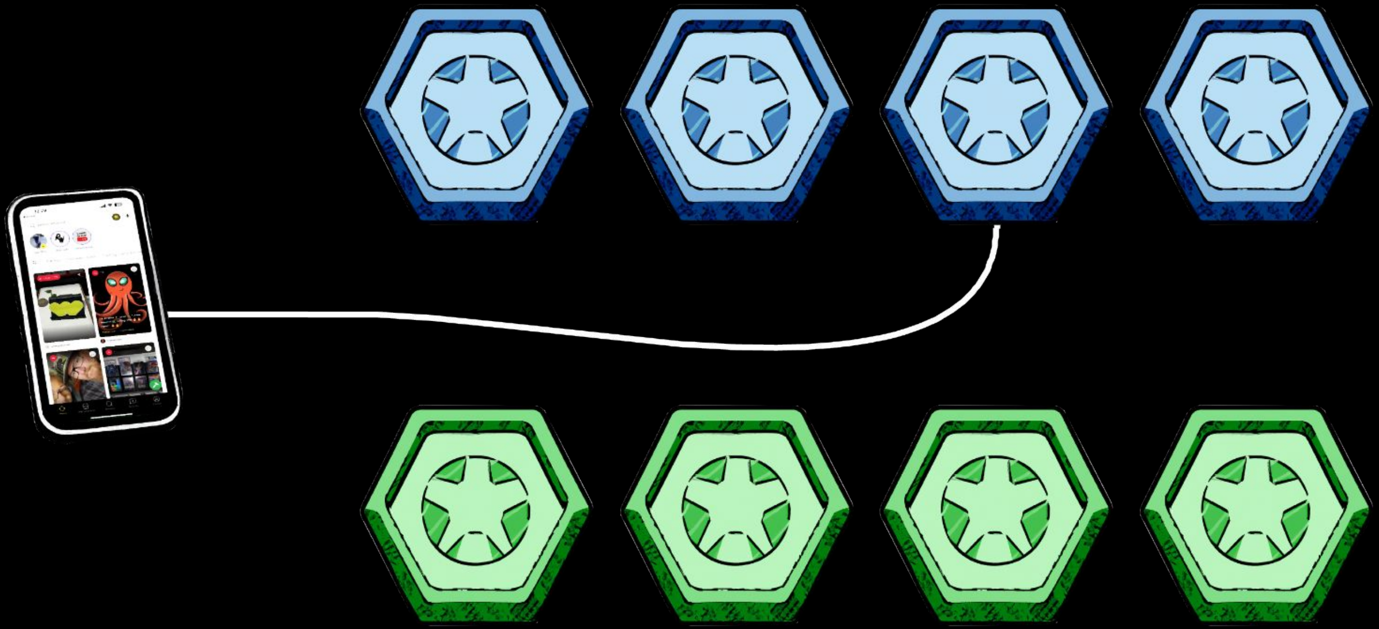
What's Changed

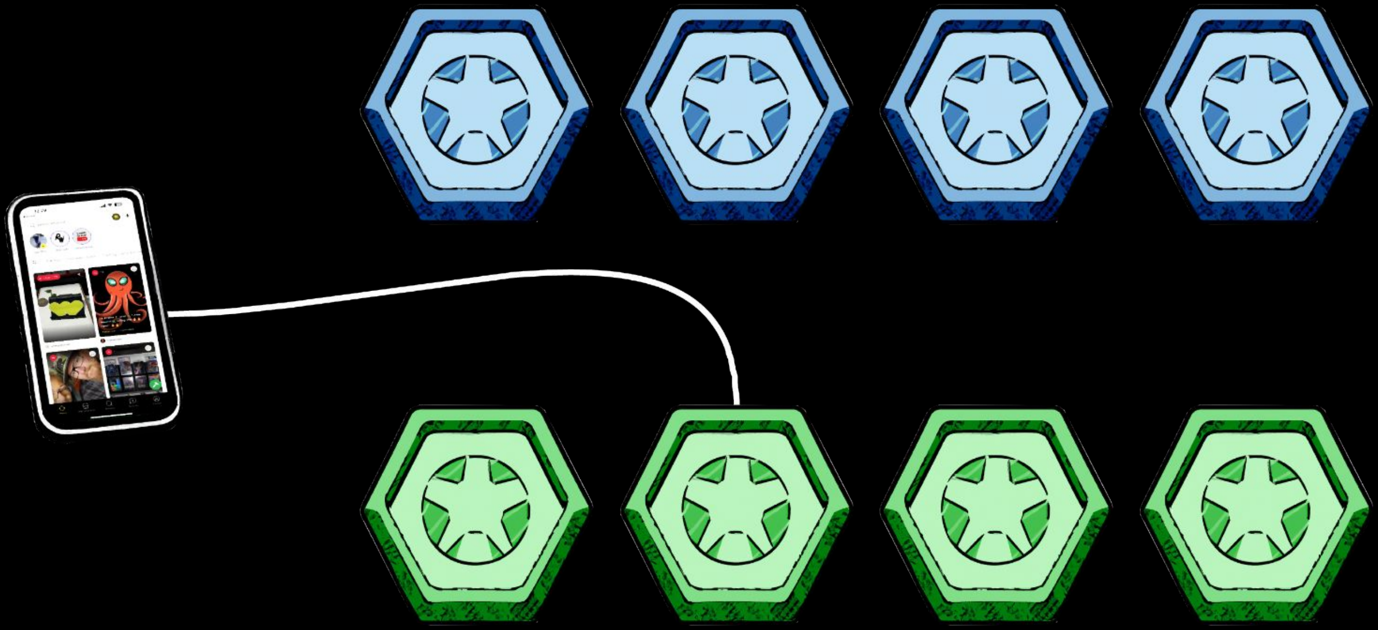
- [Bug Fix] Fixed an issue with login validation by **@johndoe** in #45
- [Enhancement] Added new feature for image cropping by **@janedoe** in #47
- [Documentation] Updated user guide with new installation instructions by **@sarahsmith** in #50

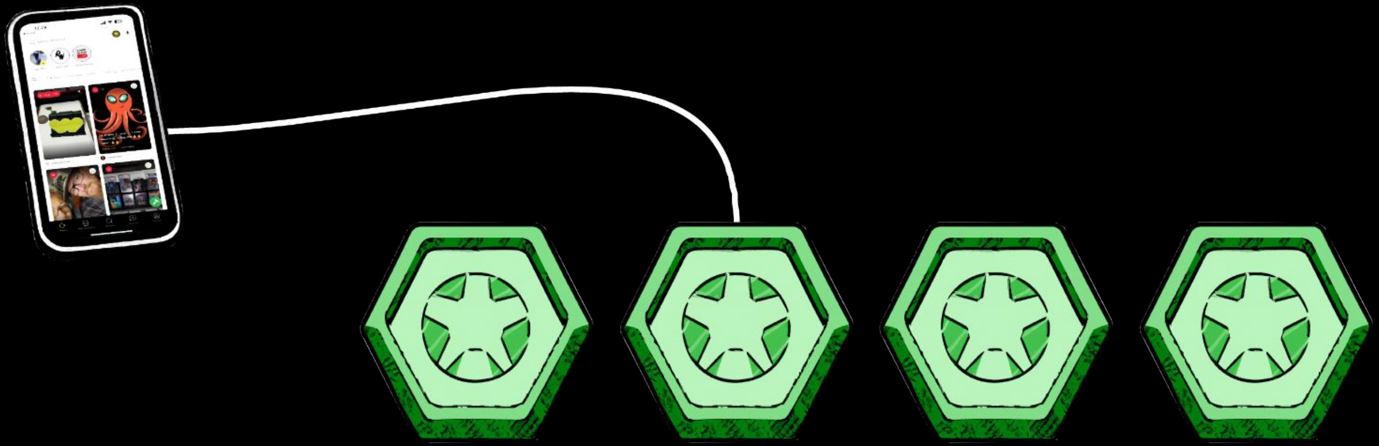
Full Changelog: `v1.0.0...v1.0.1`

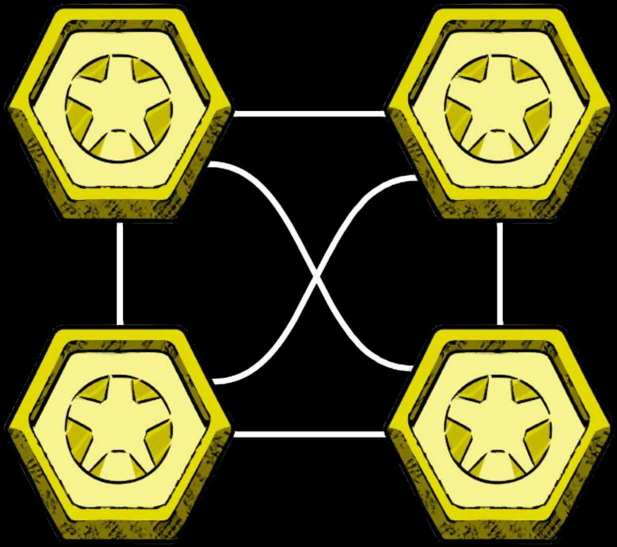

Main Backend
Live Service
Web

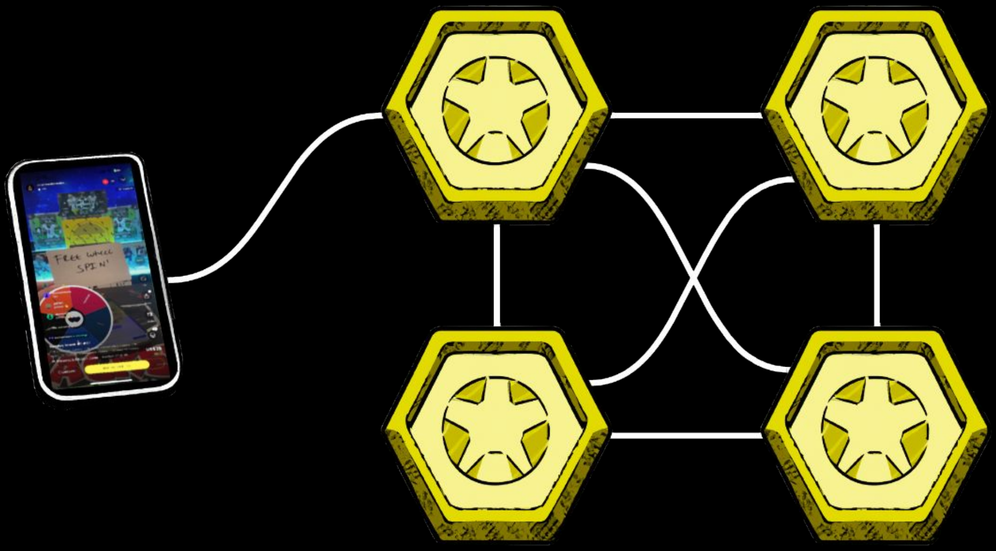


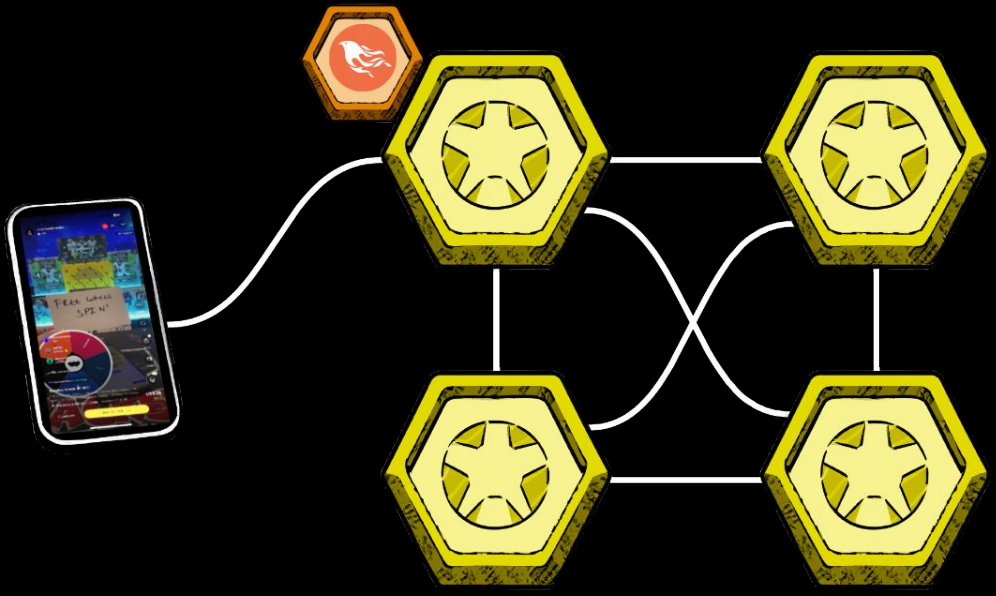


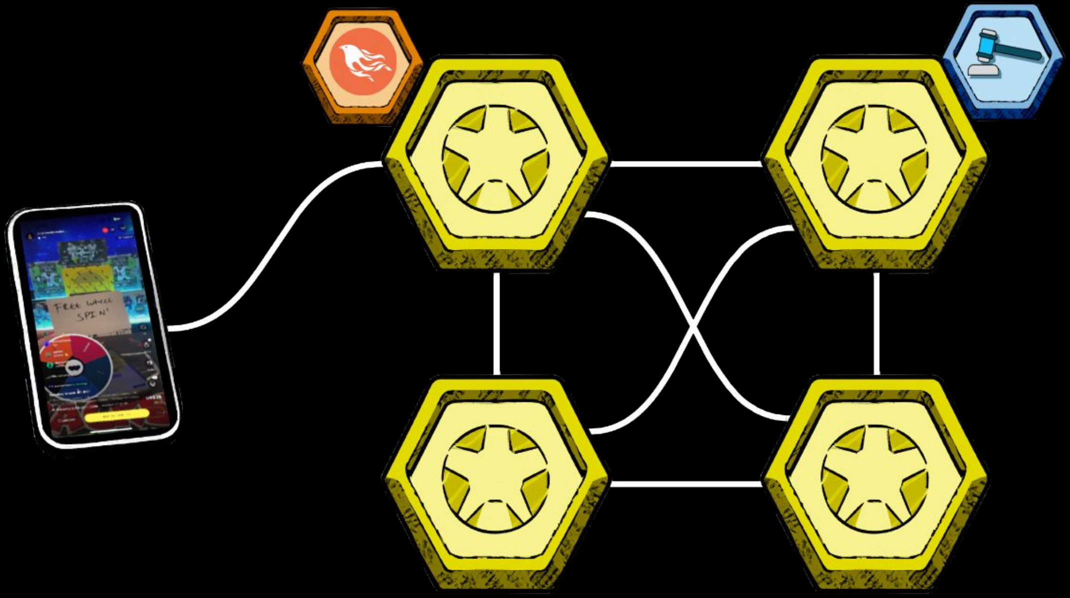


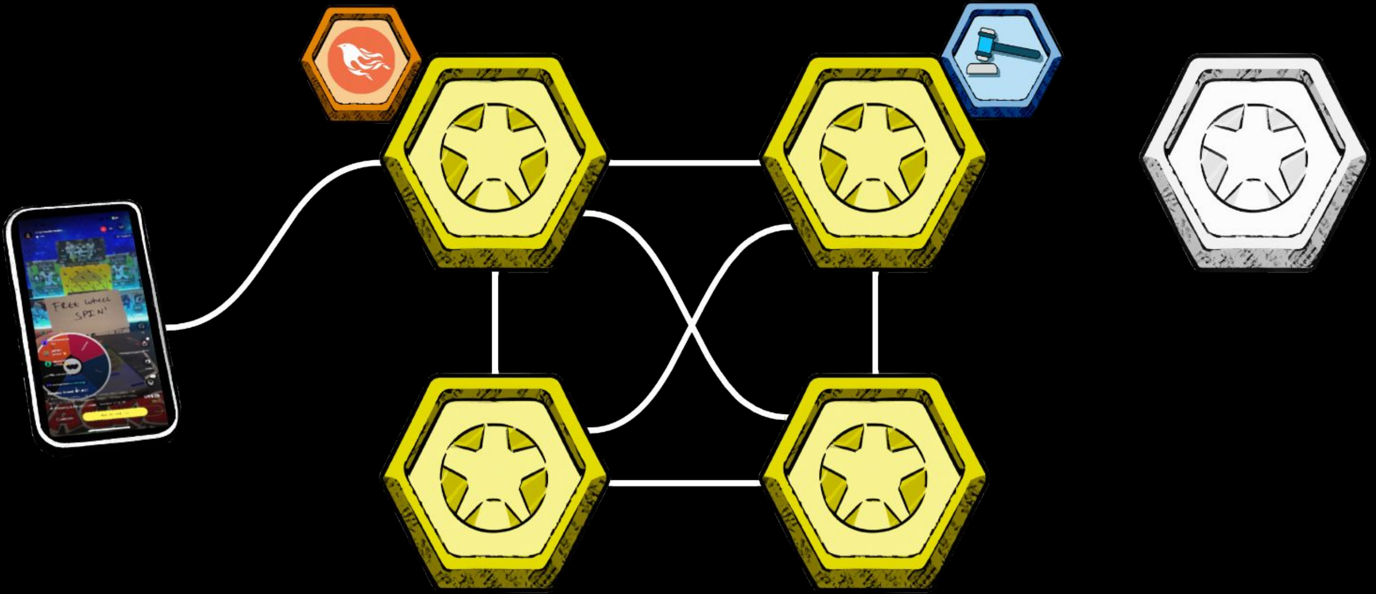


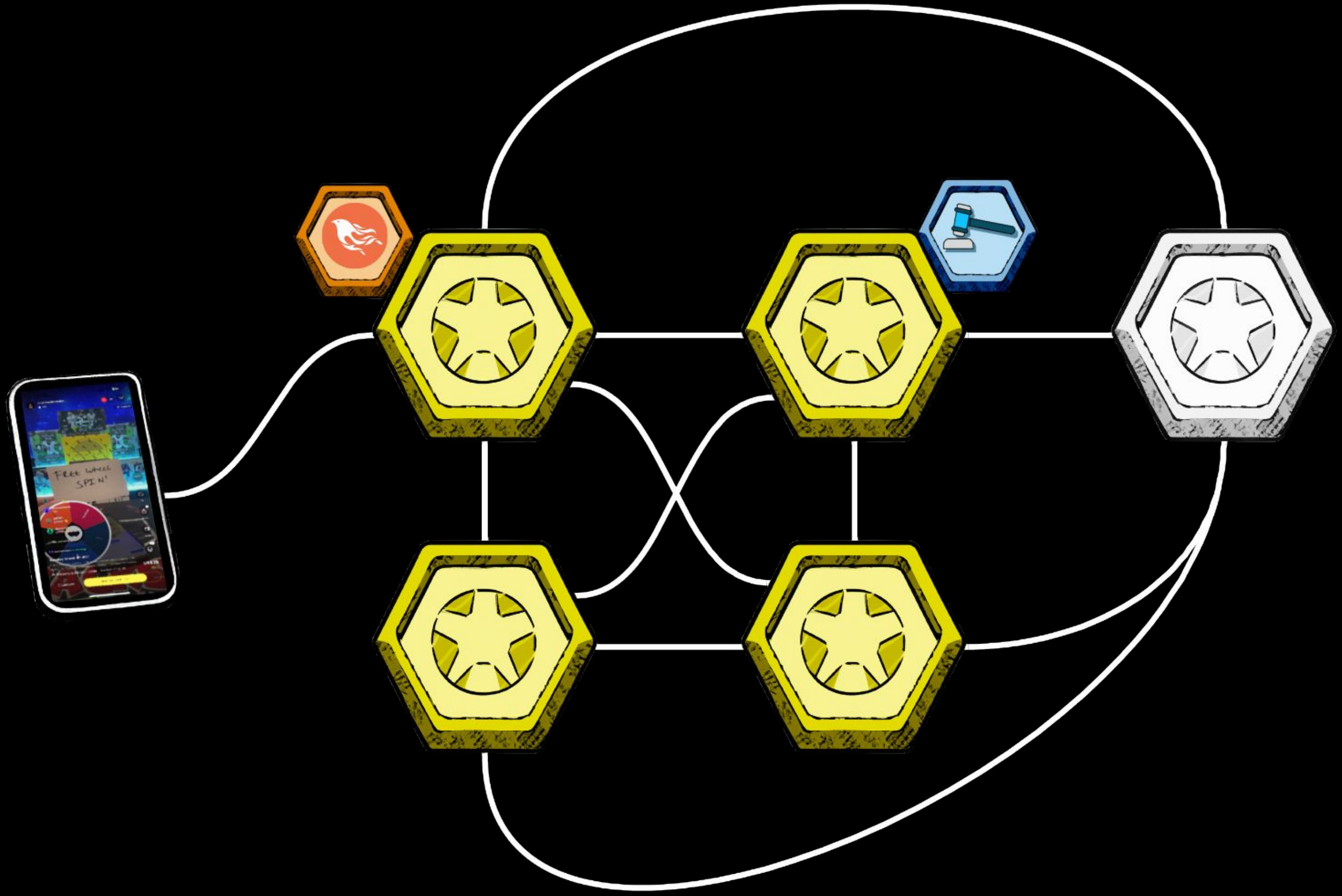


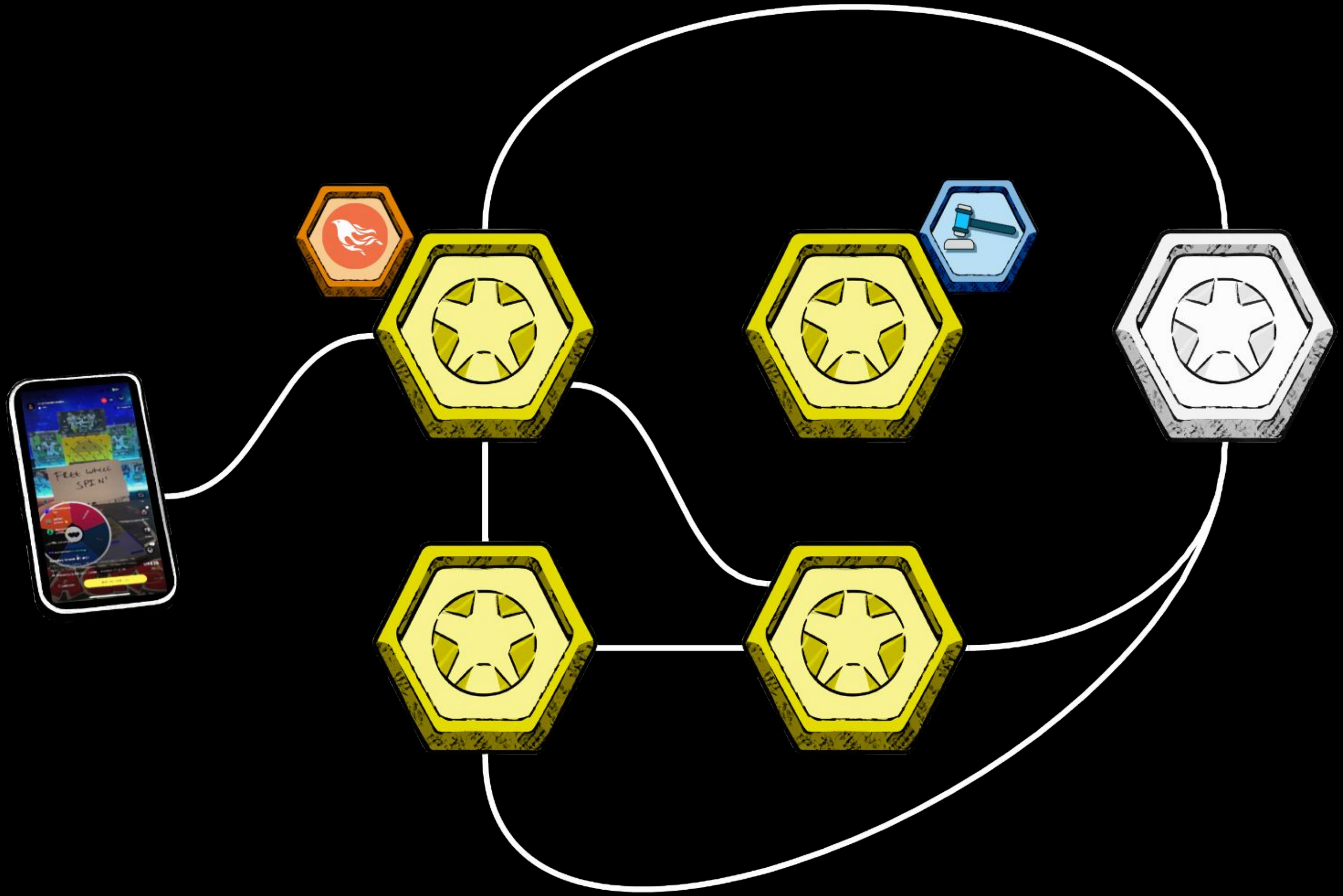

FREE WHEEL SPIN!

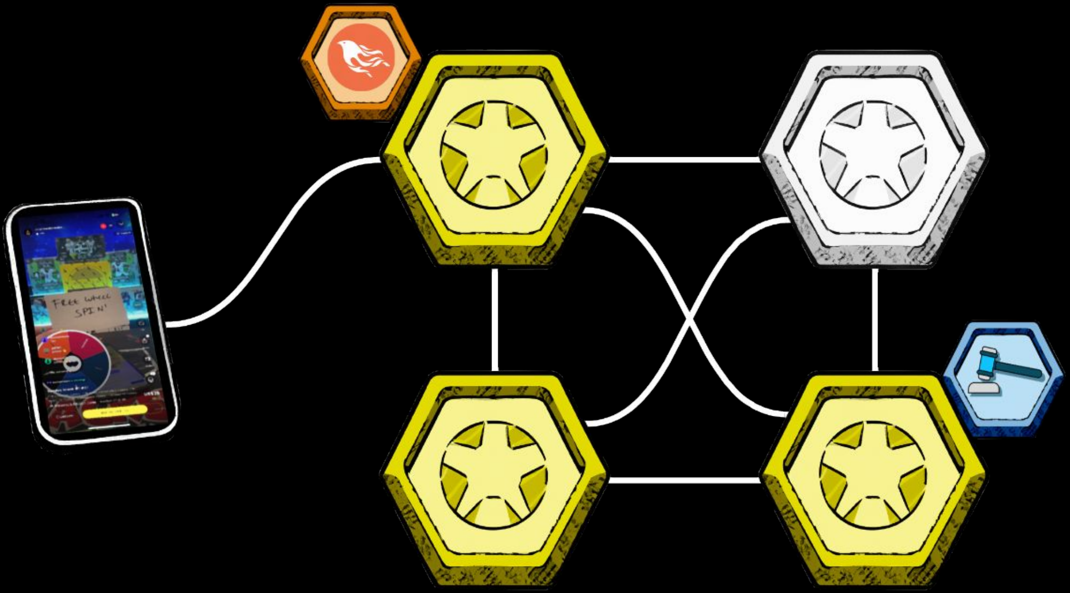

FREE WHEEL SPIN'

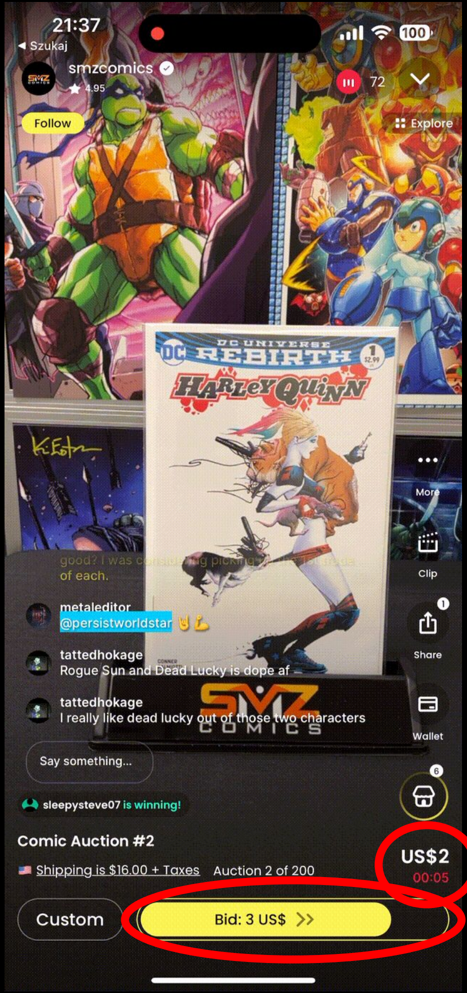

FREE WHEEL SPIN!


Free wheel SPIN!


FREE WHEEL SPIN'


FREE WHEEL SPIN'


Free wheel SPIN!


21:37
◀ Szukaj
smzcomics
4.95
72
Follow
Explore
More
Clip
Share
Wallet
good? I was considering picking up one of each.

metaleditor
@persistworldstar 🤘💪

tattedhokage
Rogue Sun and Dead Lucky is dope af

tattedhokage
I really like dead lucky out of those two characters

Say something...

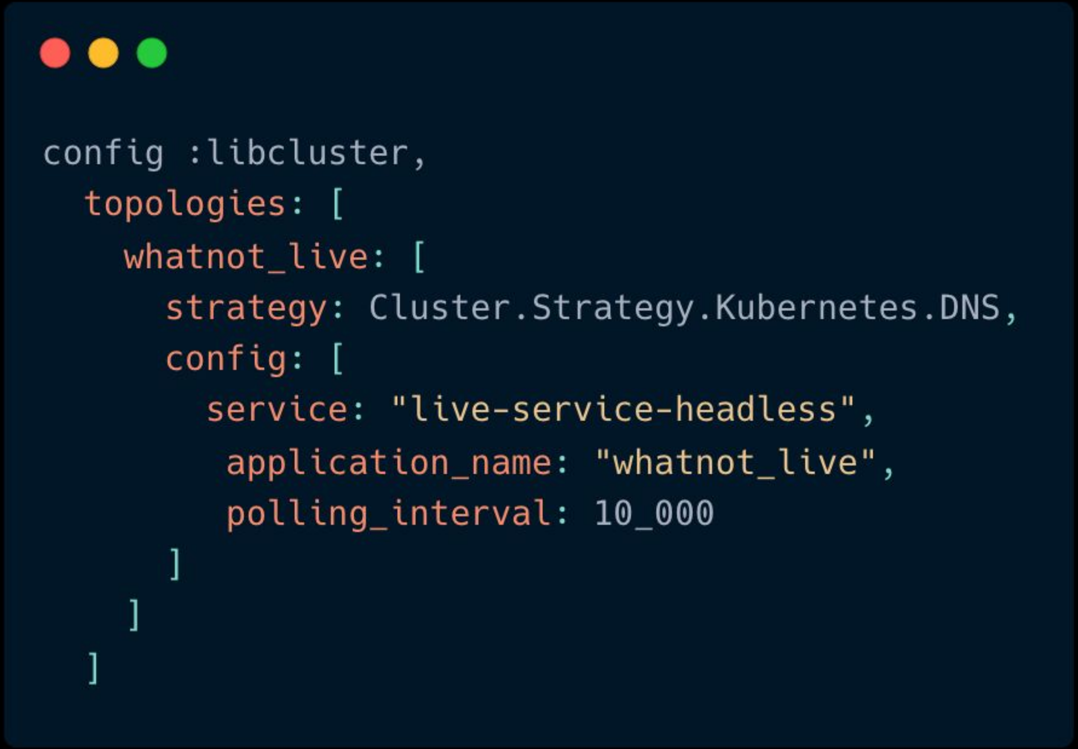
sleepysteve07 is winning!

Comic Auction #2

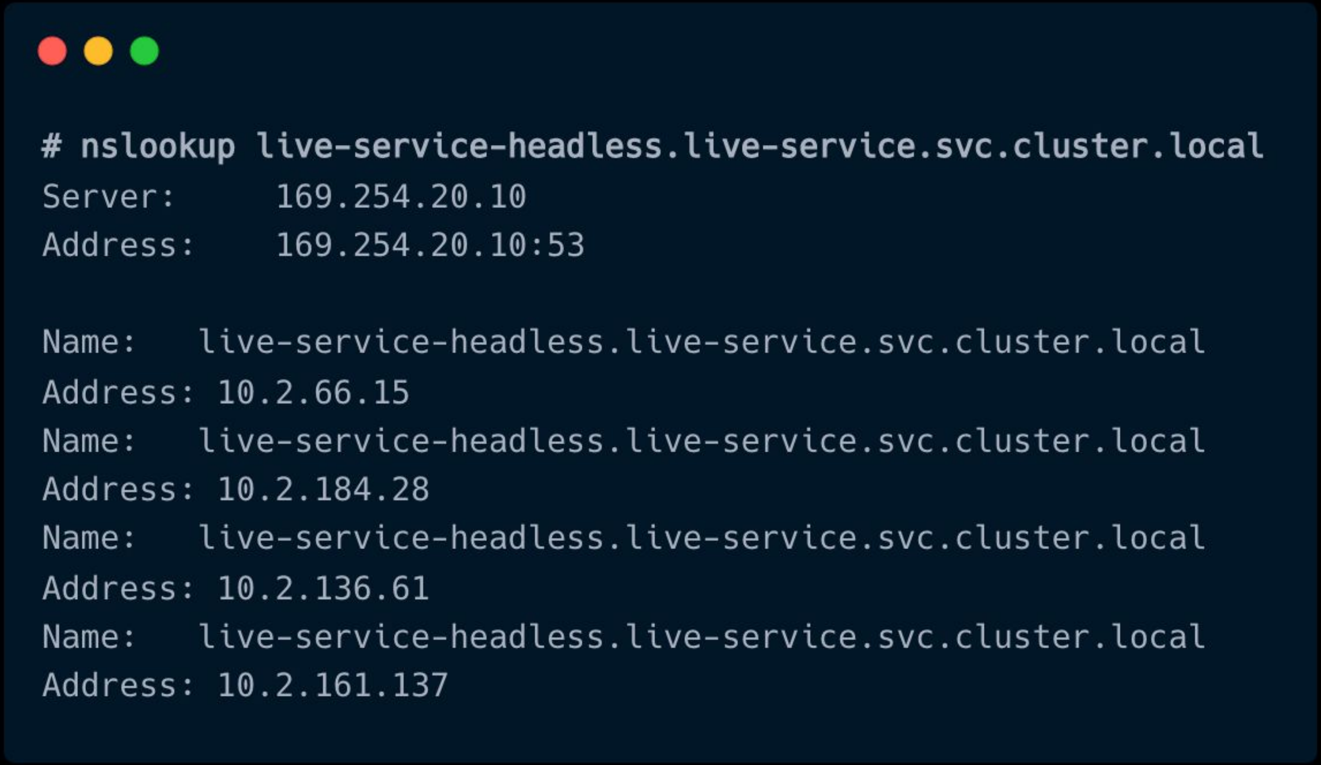
Shipping is $16.00 + Taxes   Auction 2 of 200

US$2
00:05

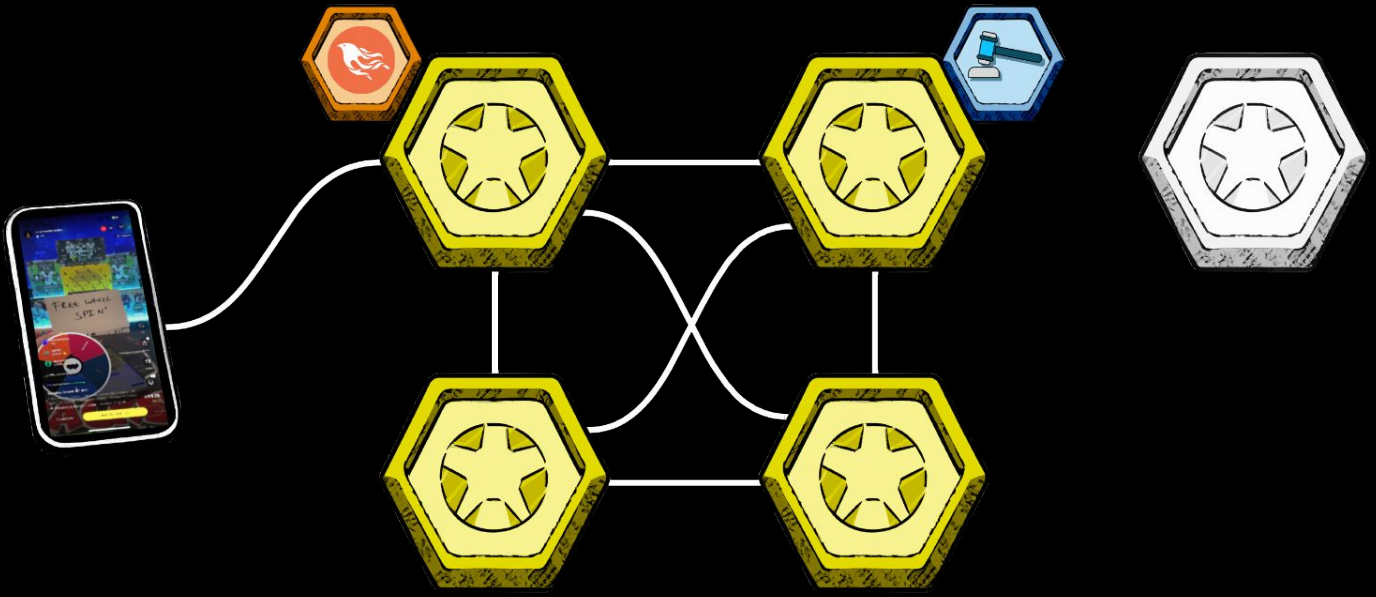
Custom   Bid: 3 US$ >>

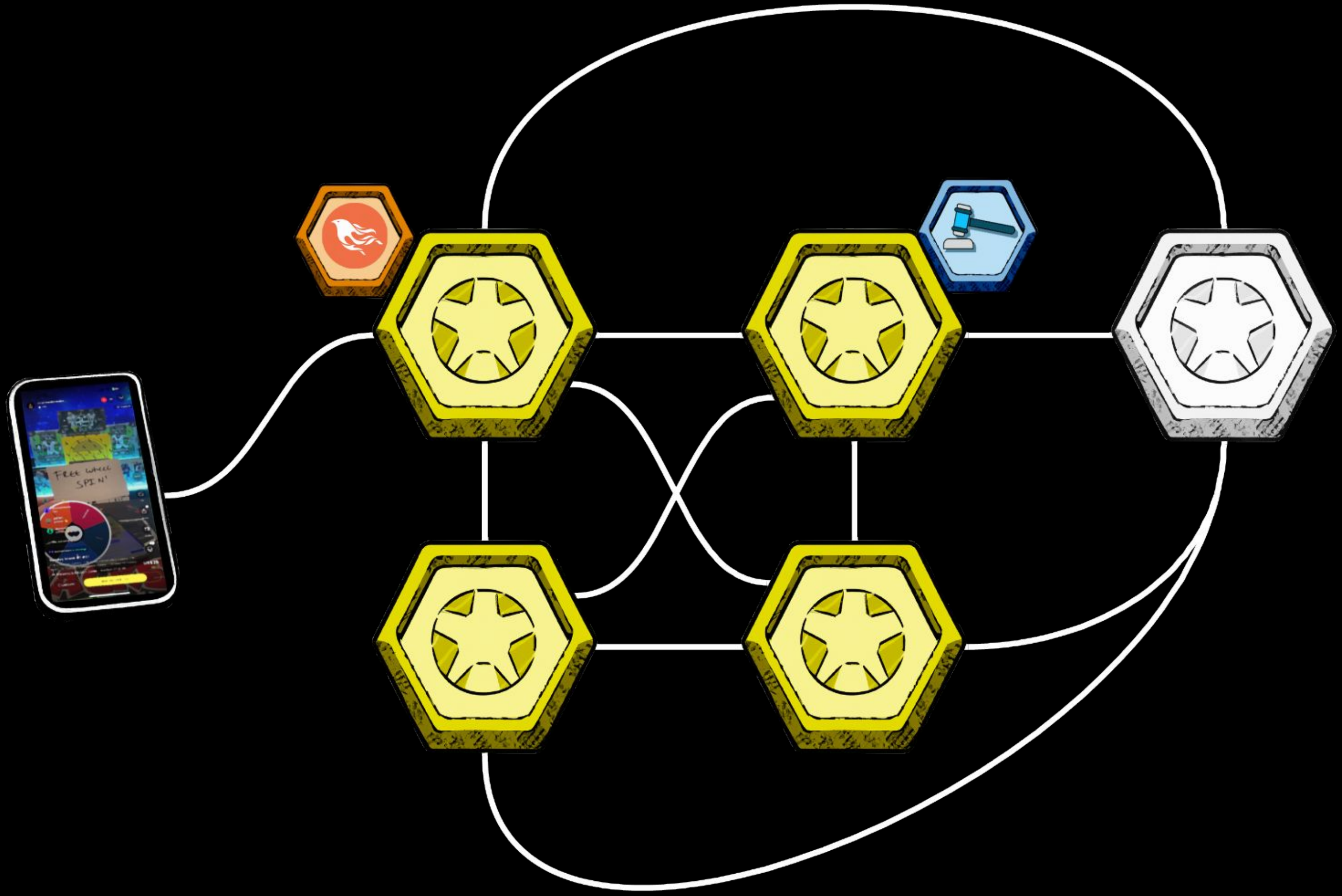


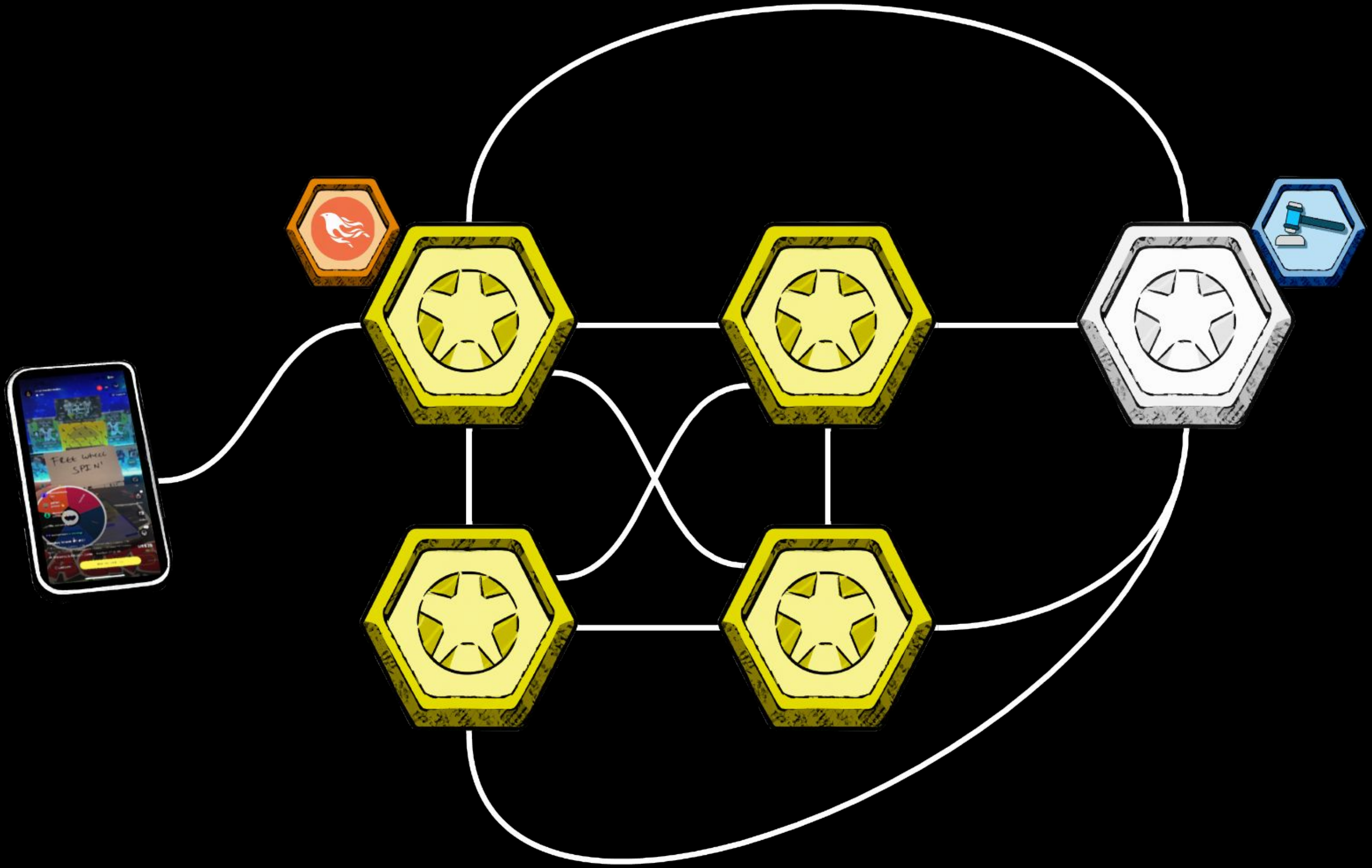
```
config :libcluster,
  topologies: [
    whatnot_live: [
      strategy: Cluster.Strategy.Kubernetes.DNS,
      config: [
        service: "live-service-headless",
         application_name: "whatnot_live",
         polling_interval: 10_000
      ]
    ]
  ]
```

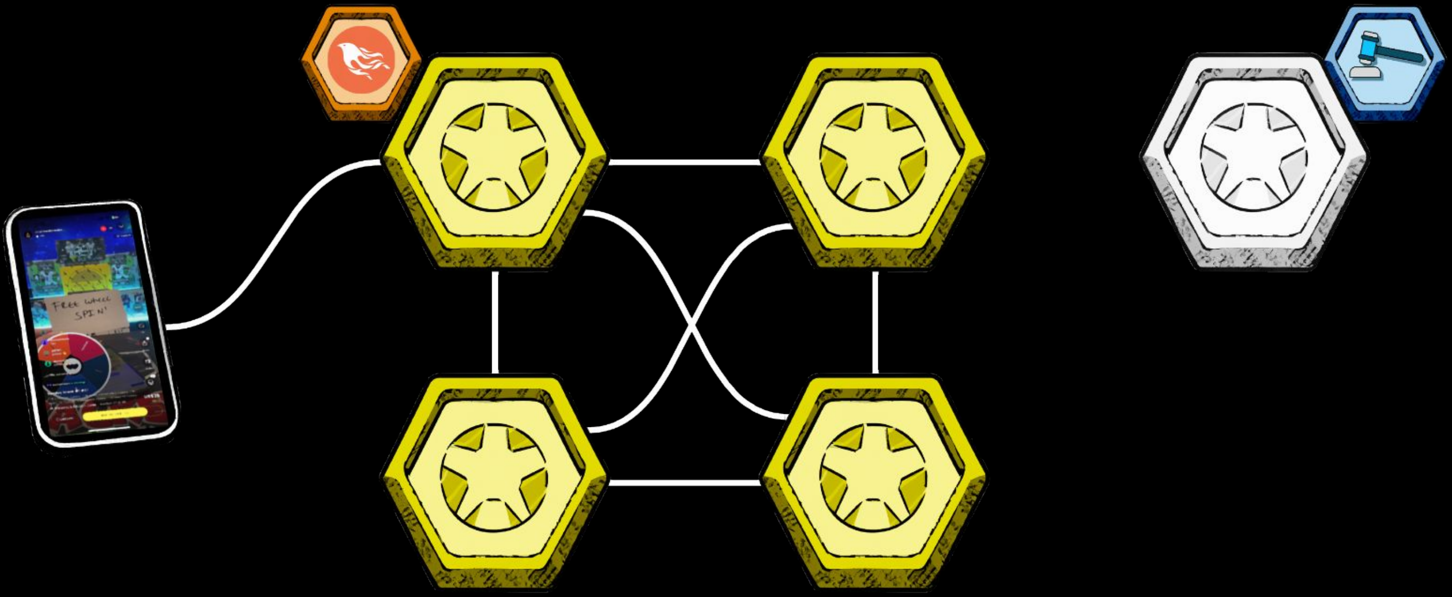


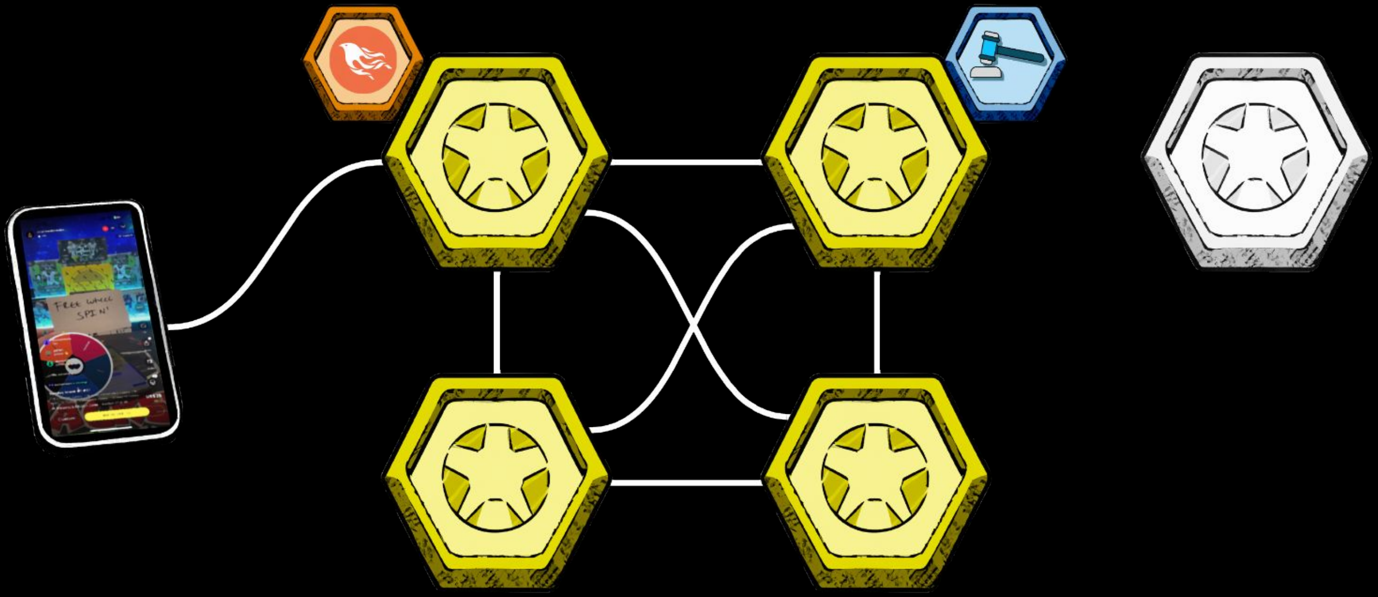
```
# nslookup live-service-headless.live-service.svc.cluster.local
Server:     169.254.20.10
Address:    169.254.20.10:53

Name:   live-service-headless.live-service.svc.cluster.local
Address: 10.2.66.15
Name:   live-service-headless.live-service.svc.cluster.local
Address: 10.2.184.28
Name:   live-service-headless.live-service.svc.cluster.local
Address: 10.2.136.61
Name:   live-service-headless.live-service.svc.cluster.local
Address: 10.2.161.137
```

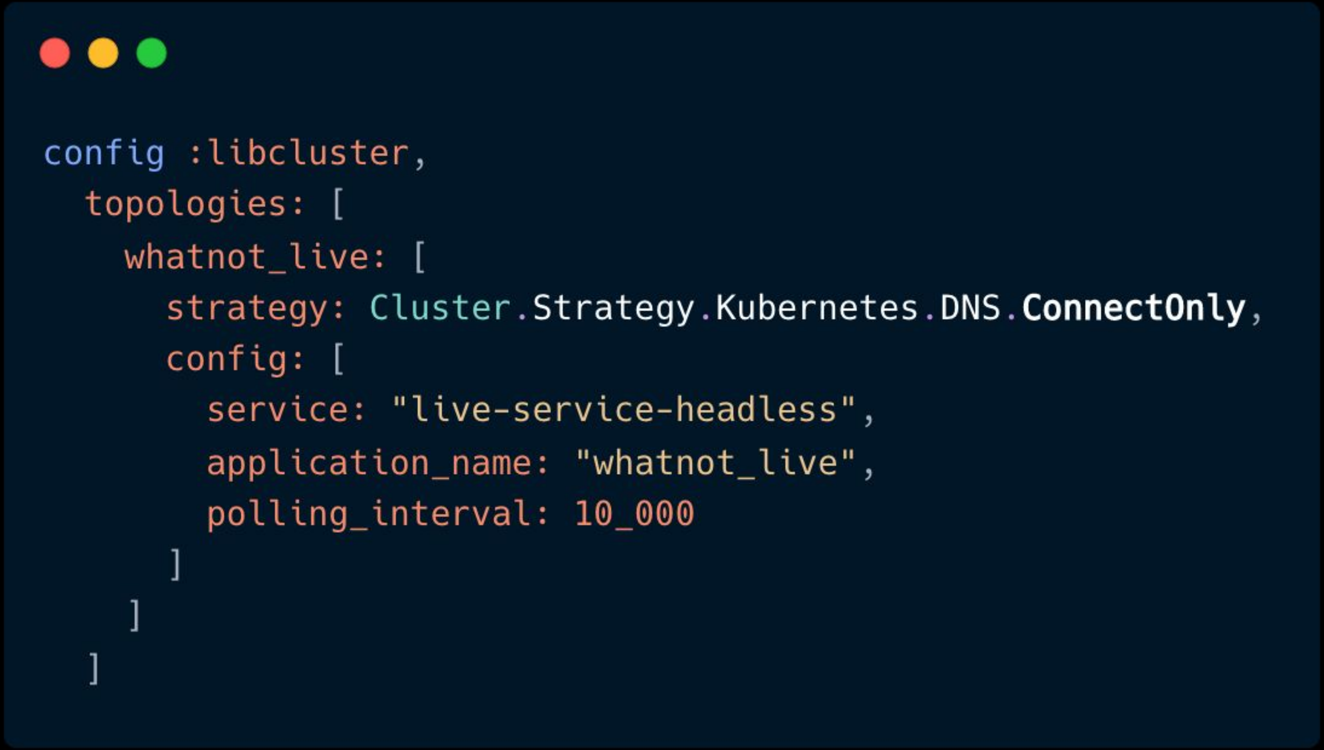

FREE WHEEL SPIN!


FREE WHEEL SPIN'


FREE WHEEL SPIN!


FREE WHEEL SPIN!

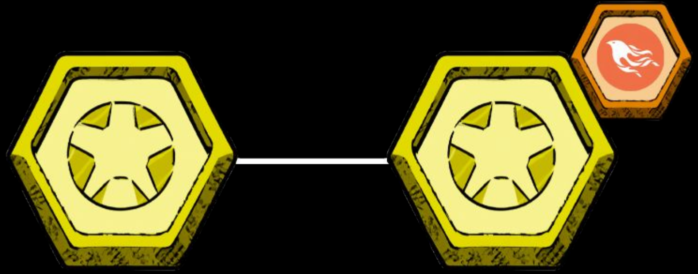

FREE WHEEL SPIN!

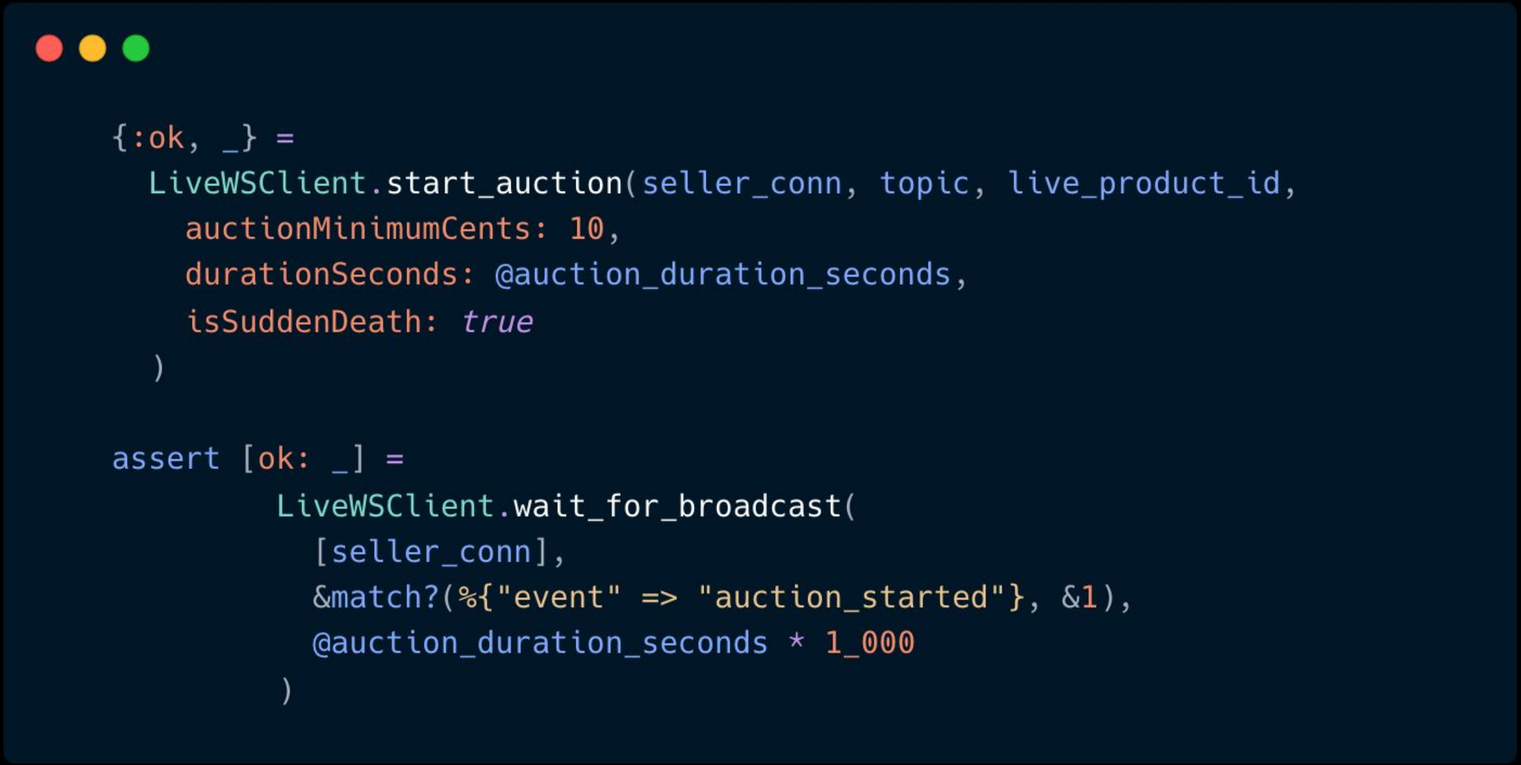


```elixir
config :libcluster,
  topologies: [
    whatnot_live: [
      strategy: Cluster.Strategy.Kubernetes.DNS.ConnectOnly,
      config: [
        service: "live-service-headless",
        application_name: "whatnot_live",
        polling_interval: 10_000
      ]
    ]
  ]
```

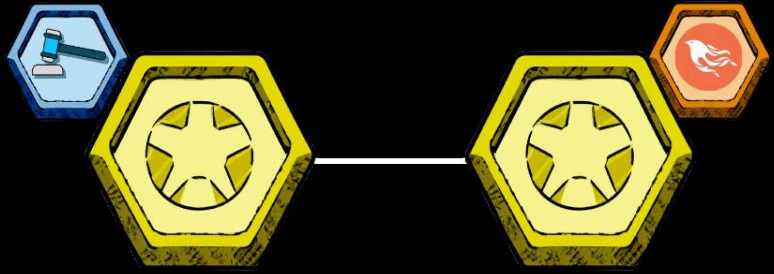
```elixir
setup_all do
  current_version = System.get_env("TAG", "latest")
  # last tag:
  previous_version = System.get_env("PREVIOUS_TAG", previous_version())

  [
    {current_version, "HEAD"},
    {previous_version, previous_version}
  ]
  |> Task.async_stream(fn {tag, ref} -> ensure_image_present(tag, ref) end,
    timeout: :infinity
  )
  |> Stream.run()

  [current_version: current_version, previous_version: previous_version]
end
```

```elixir
test "Users can bid during version upgrade", %{
  current_version: current_version,
  previous_version: previous_version
} do
  import UserGivens
  import AuctionGivens

  local_cluster_stopped()
  node_running(14441, tag: previous_version)
  node_running(14442, tag: previous_version)
  [{_, seller_conn}] = users = given_users_connected(1, port: 14442)

  %{
    topic: topic,
    livestream_id: livestream_id,
    live_product_id: live_product_id
  } = given_livestream_with_users(users, %{}, port: 14442)
```

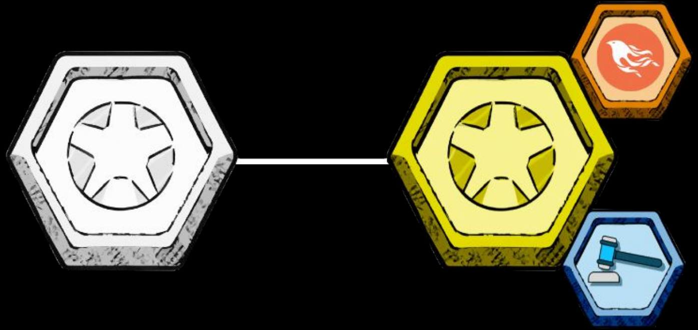


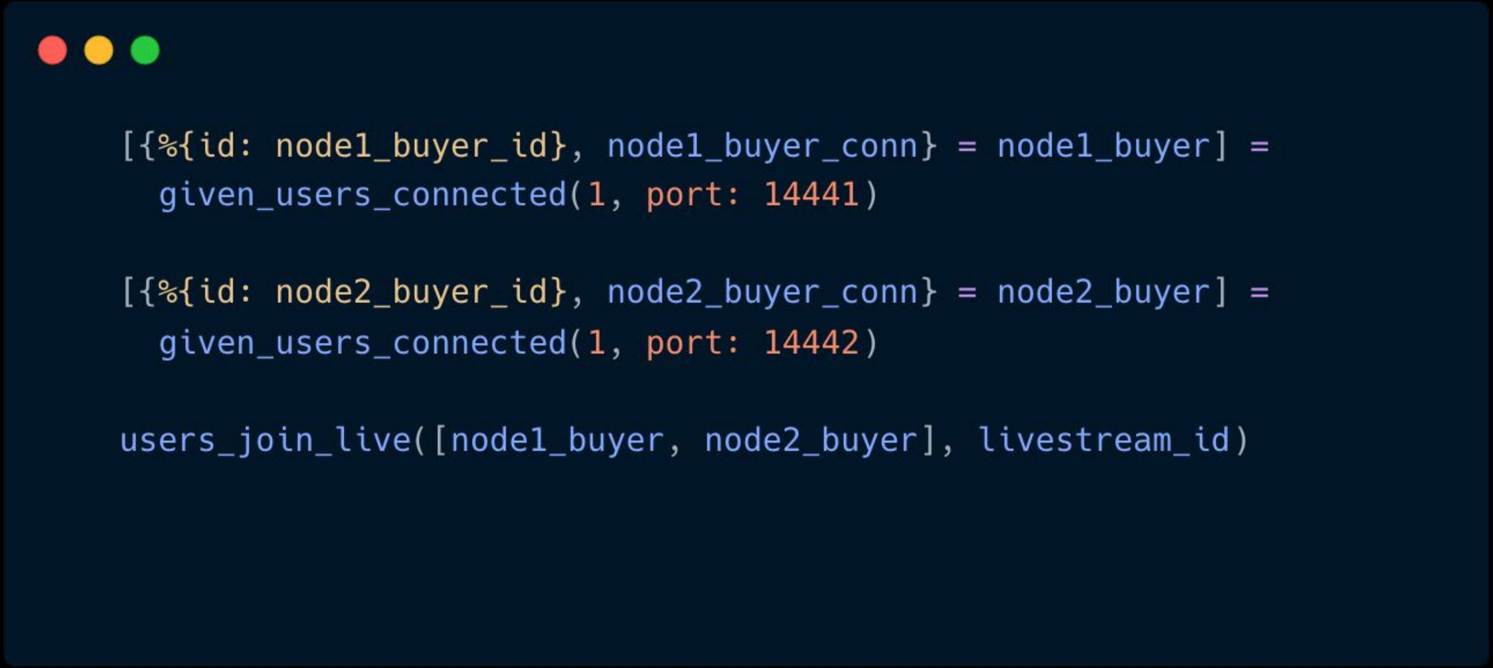


```
{:ok, _} =
  LiveWSClient.start_auction(seller_conn, topic, live_product_id,
    auctionMinimumCents: 10,
    durationSeconds: @auction_duration_seconds,
    isSuddenDeath: true
  )

assert [ok: _] =
         LiveWSClient.wait_for_broadcast(
           [seller_conn],
           &match?(%{"event" => "auction_started"}, &1),
           @auction_duration_seconds * 1_000
         )
```

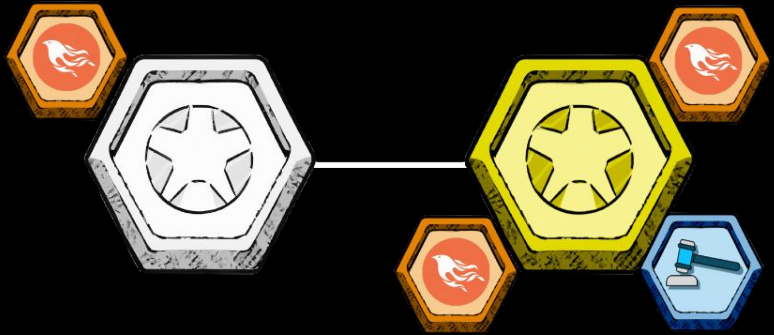


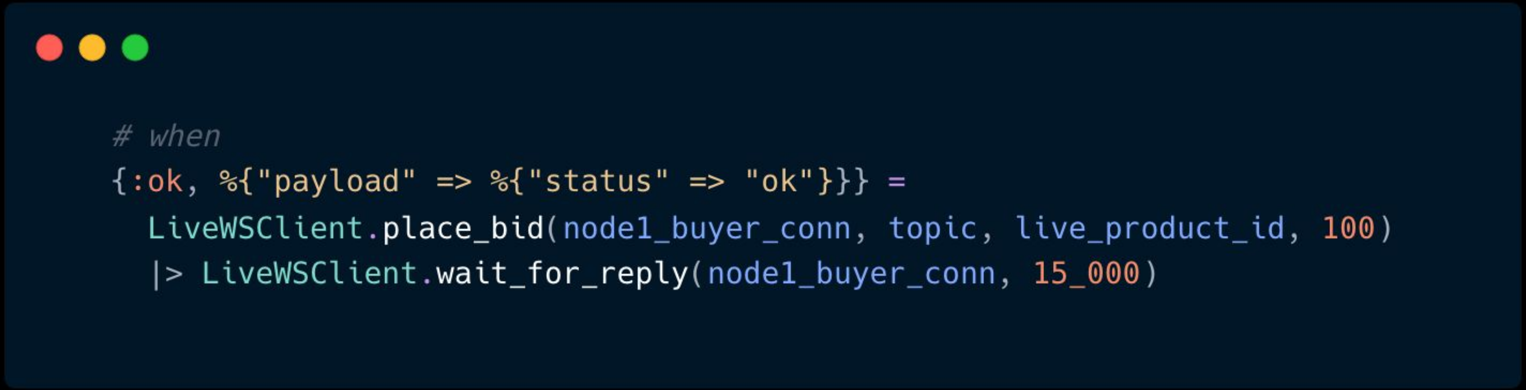
```elixir
node_stopped(14441)
node_running(14441, tag: current_version)

eventually(
  fn ->
    assert nodes_connected_to(14441) == 1
    assert nodes_connected_to(14442) == 1
  end,
  30,
  1_000
)
```



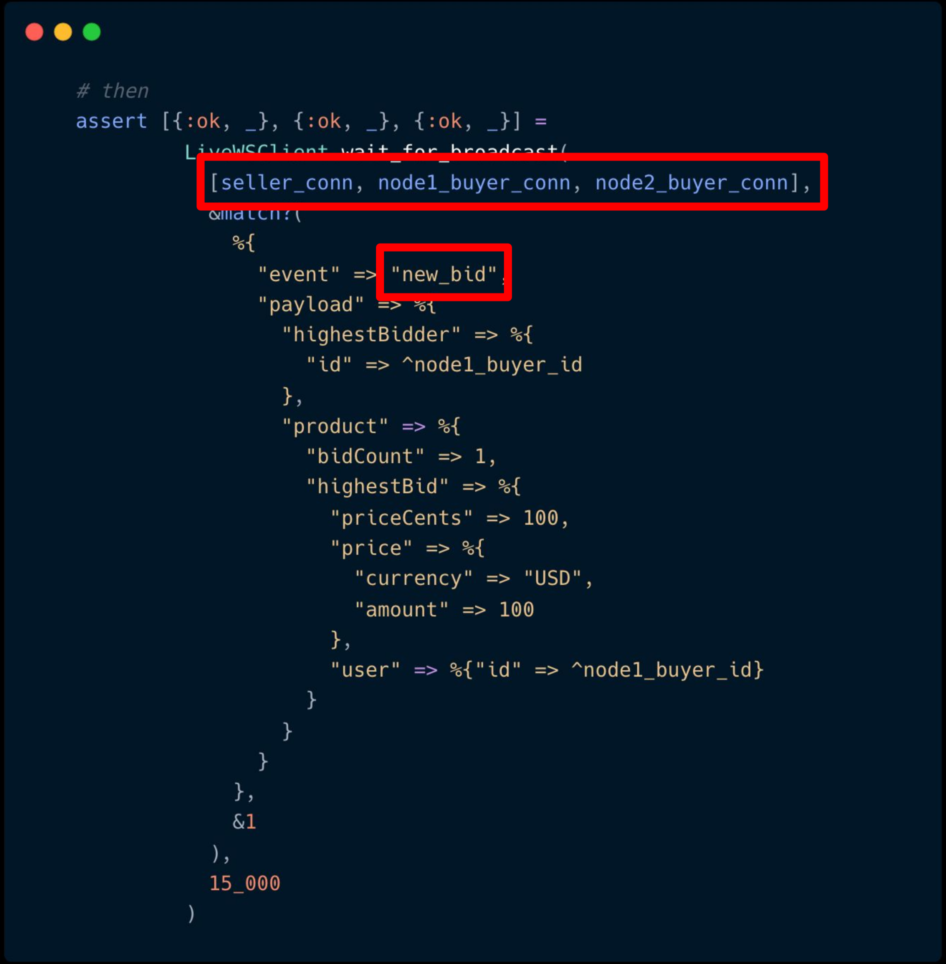


```elixir
[{%{id: node1_buyer_id}, node1_buyer_conn} = node1_buyer] =
  given_users_connected(1, port: 14441)

[{%{id: node2_buyer_id}, node2_buyer_conn} = node2_buyer] =
  given_users_connected(1, port: 14442)

users_join_live([node1_buyer, node2_buyer], livestream_id)
```

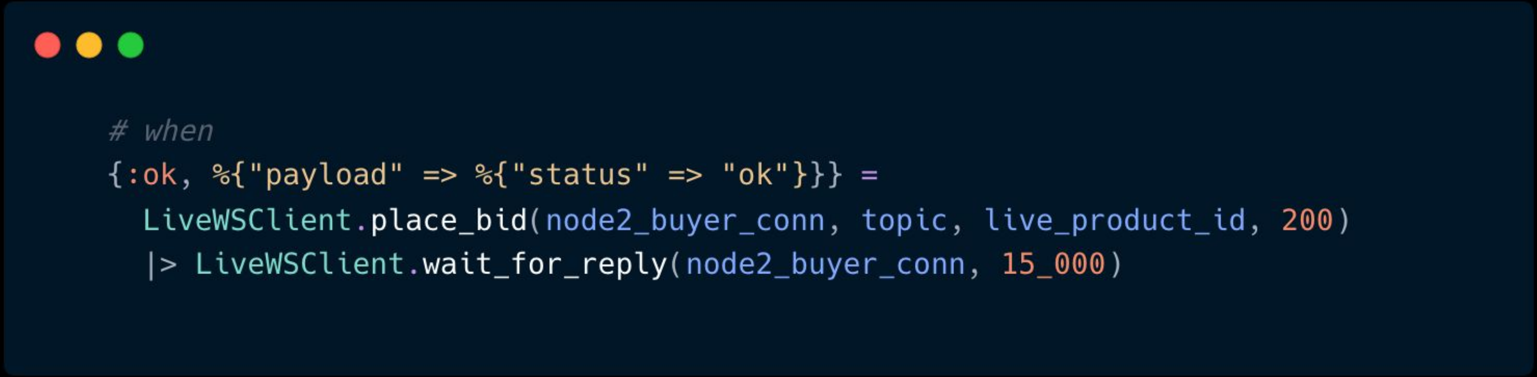




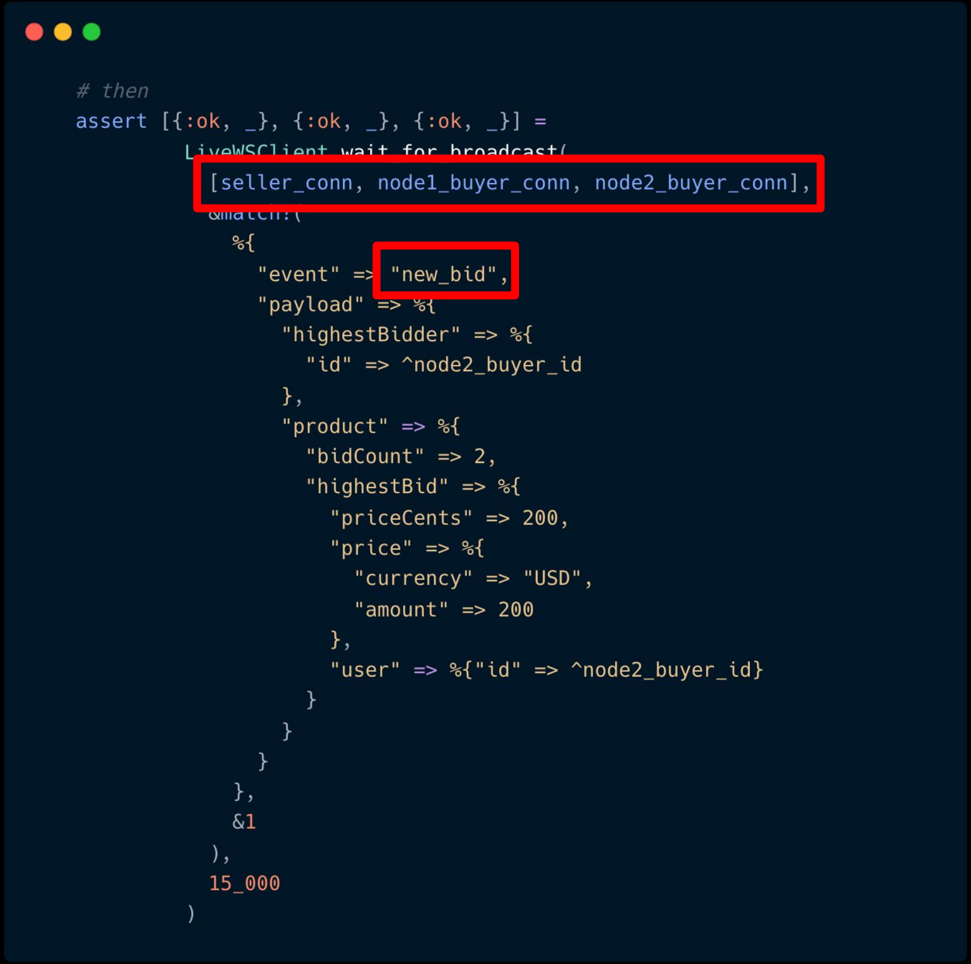
```elixir
# when
{:ok, %{"payload" => %{"status" => "ok"}}} =
  LiveWSClient.place_bid(node1_buyer_conn, topic, live_product_id, 100)
  |> LiveWSClient.wait_for_reply(node1_buyer_conn, 15_000)
```



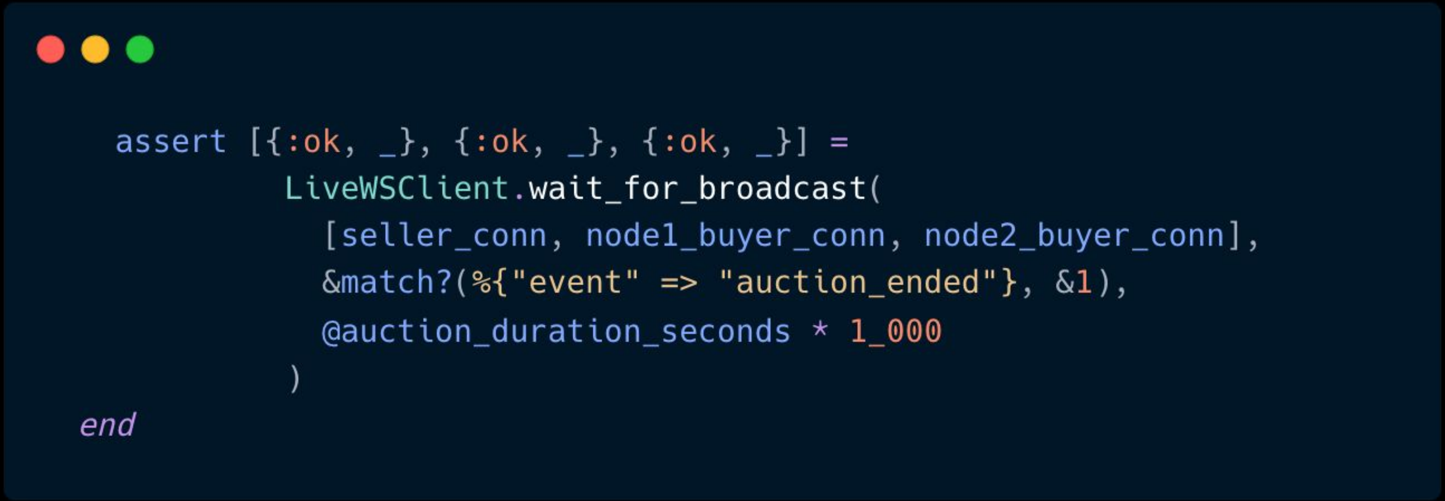
```
# then
assert [{:ok, _}, {:ok, _}, {:ok, _}] =
         LiveWSClient.wait_for_broadcast(
           [seller_conn, node1_buyer_conn, node2_buyer_conn],
           &match?(
             %{
               "event" => "new_bid",
               "payload" => %{
                 "highestBidder" => %{
                   "id" => ^node1_buyer_id
                 },
                 "product" => %{
                   "bidCount" => 1,
                   "highestBid" => %{
                     "priceCents" => 100,
                     "price" => %{
                       "currency" => "USD",
                       "amount" => 100
                     },
                     "user" => %{"id" => ^node1_buyer_id}
                   }
                 }
               }
             },
             &1
           ),
           15_000
         )
```



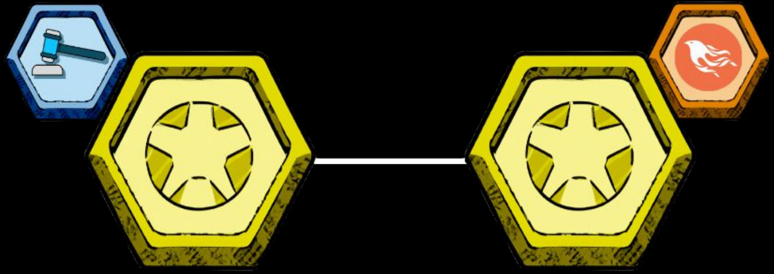
```elixir
# when
{:ok, %{"payload" => %{"status" => "ok"}}} =
  LiveWSClient.place_bid(node2_buyer_conn, topic, live_product_id, 200)
  |> LiveWSClient.wait_for_reply(node2_buyer_conn, 15_000)
```

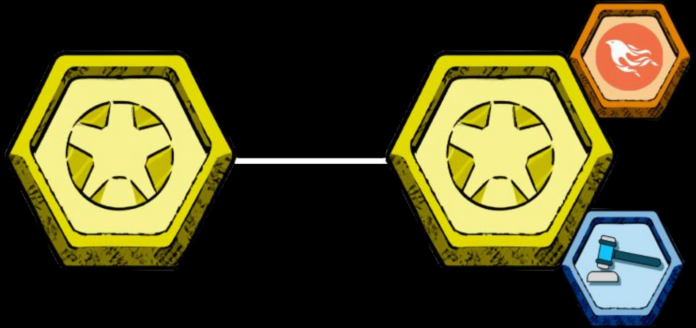


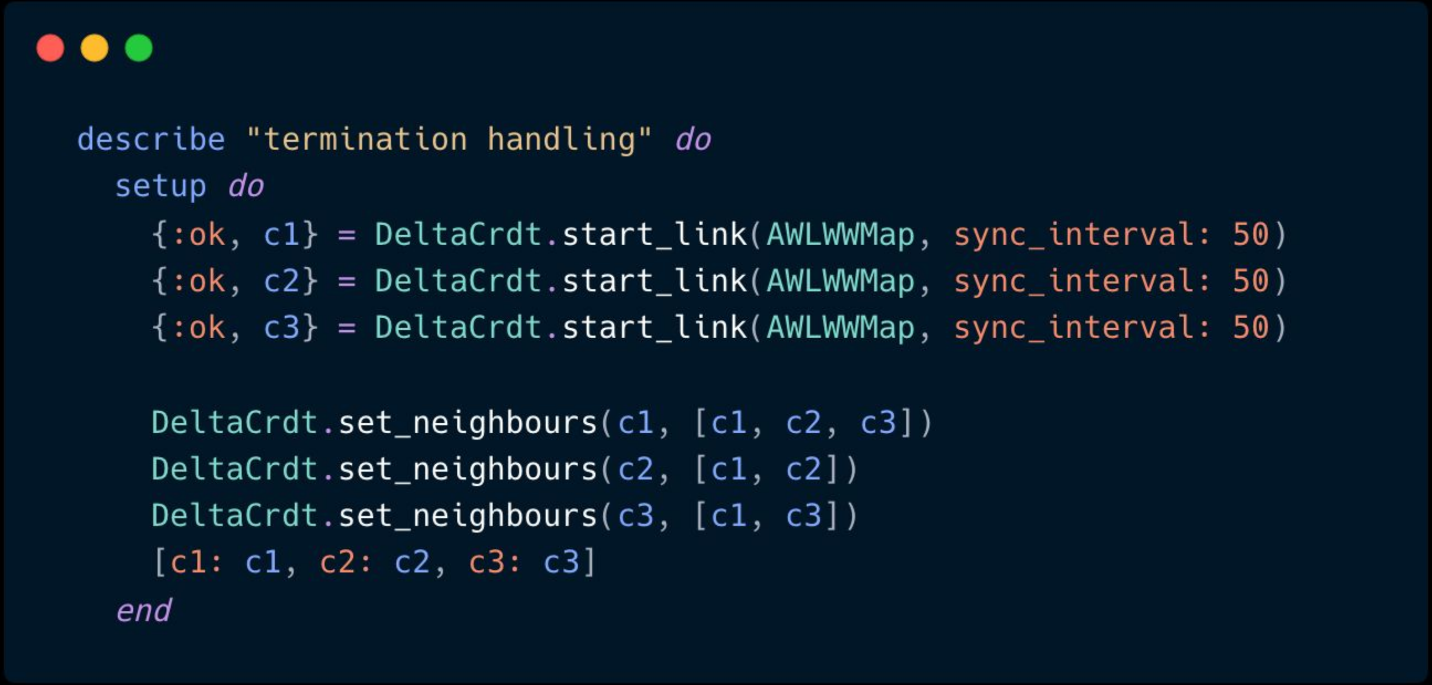
```elixir
# then
assert [{:ok, _}, {:ok, _}, {:ok, _}] =
         LiveWSClient.wait_for_broadcast(
           [seller_conn, node1_buyer_conn, node2_buyer_conn],
           &match?(
             %{
               "event" => "new_bid",
               "payload" => %{
                 "highestBidder" => %{
                   "id" => ^node2_buyer_id
                 },
                 "product" => %{
                   "bidCount" => 2,
                   "highestBid" => %{
                     "priceCents" => 200,
                     "price" => %{
                       "currency" => "USD",
                       "amount" => 200
                     },
                     "user" => %{"id" => ^node2_buyer_id}
                   }
                 }
               }
             },
             &1
           ),
           15_000
         )
```



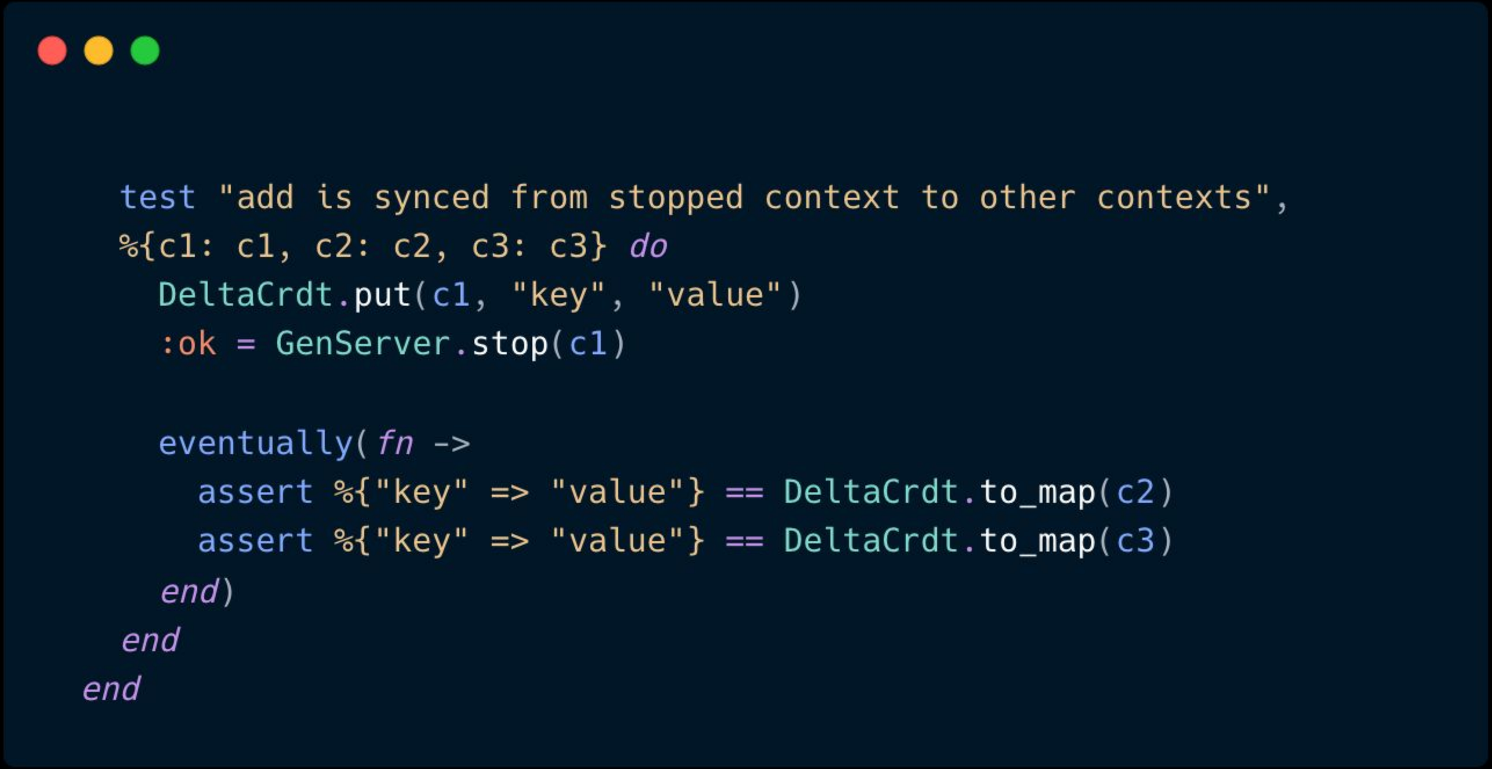
```
    assert [{:ok, _}, {:ok, _}, {:ok, _}] =
             LiveWSClient.wait_for_broadcast(
               [seller_conn, node1_buyer_conn, node2_buyer_conn],
               &match?(%{"event" => "auction_ended"}, &1),
               @auction_duration_seconds * 1_000
             )
  end
```





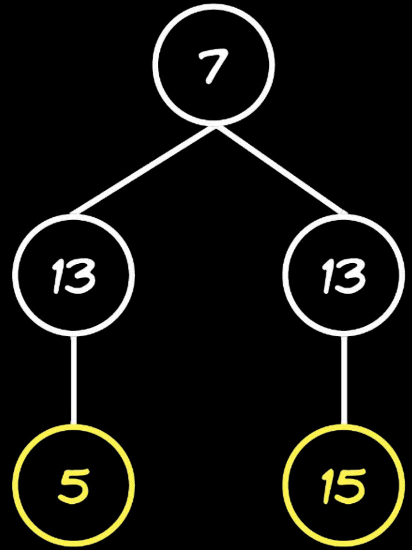


```
describe "termination handling" do
  setup do
    {:ok, c1} = DeltaCrdt.start_link(AWLWWMap, sync_interval: 50)
    {:ok, c2} = DeltaCrdt.start_link(AWLWWMap, sync_interval: 50)
    {:ok, c3} = DeltaCrdt.start_link(AWLWWMap, sync_interval: 50)

    DeltaCrdt.set_neighbours(c1, [c1, c2, c3])
    DeltaCrdt.set_neighbours(c2, [c1, c2])
    DeltaCrdt.set_neighbours(c3, [c1, c3])
    [c1: c1, c2: c2, c3: c3]
  end
```



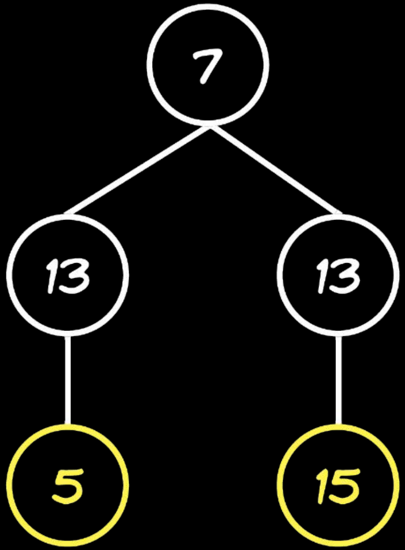
```
  test "add is synced from stopped context to other contexts",
  %{c1: c1, c2: c2, c3: c3} do
    DeltaCrdt.put(c1, "key", "value")
    :ok = GenServer.stop(c1)

    eventually(fn ->
      assert %{"key" => "value"} == DeltaCrdt.to_map(c2)
      assert %{"key" => "value"} == DeltaCrdt.to_map(c3)
    end)
  end
end
```

```
%{"a" => "a", "b" => "b"}
```



```
%{"b" => "b"}        %{"a" => "a"}
```

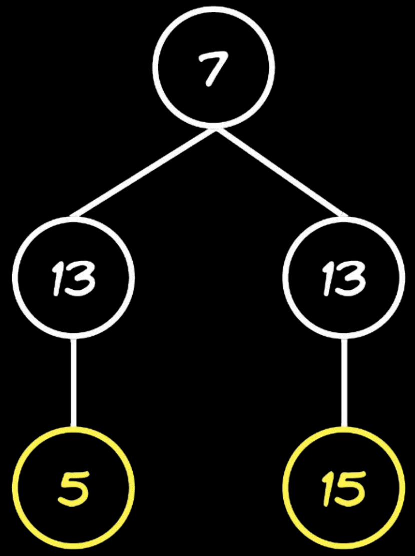
%{"a" => "a", "b" => "b"}



hash("a")

%{"b" => "b"} %{"a" => "a"}

%{"a" => "a", "b" => "b"}

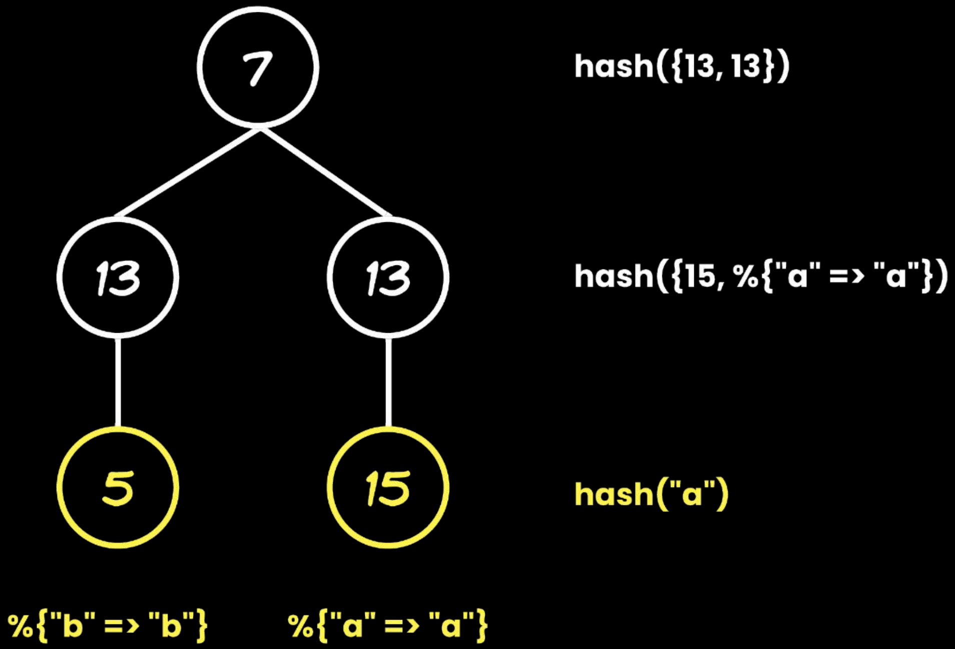


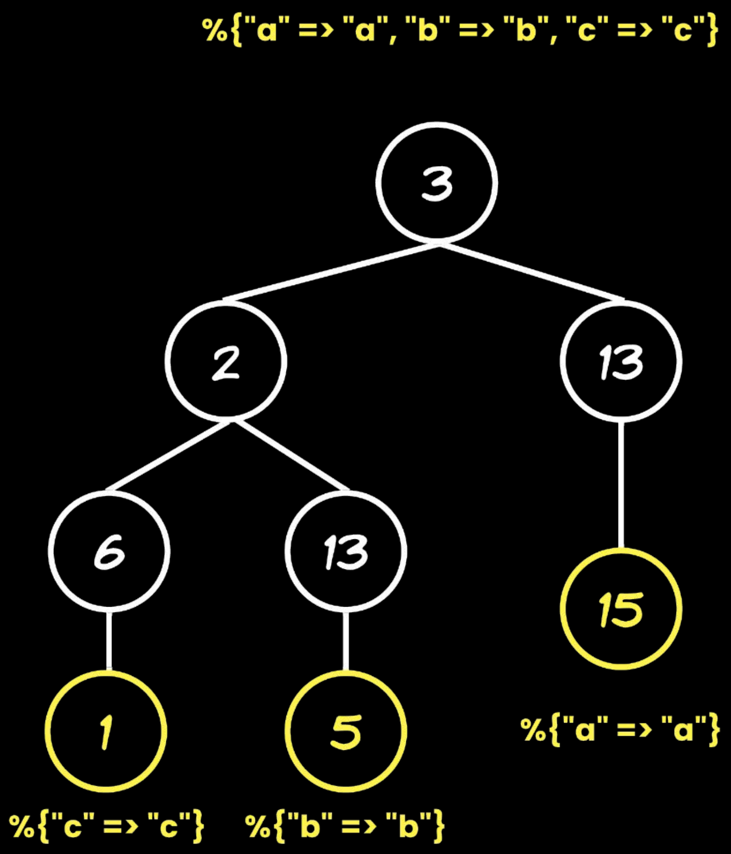
hash({15, %{"a" => "a"}})

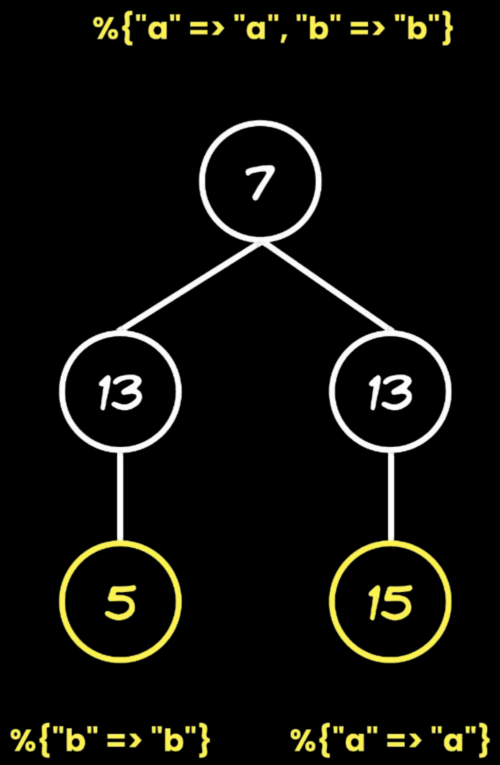
hash("a")

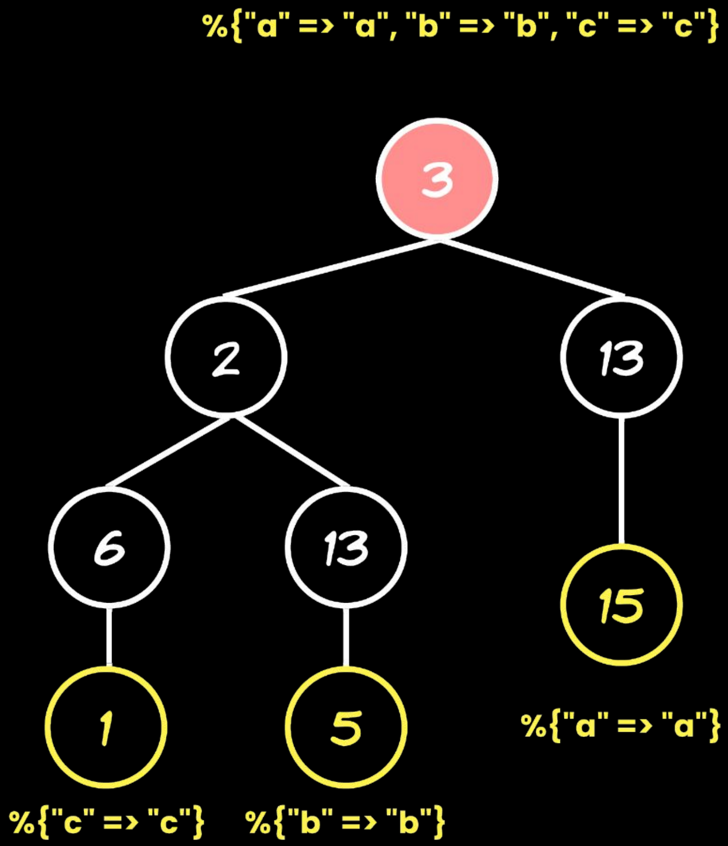
%{"b" => "b"}　　%{"a" => "a"}

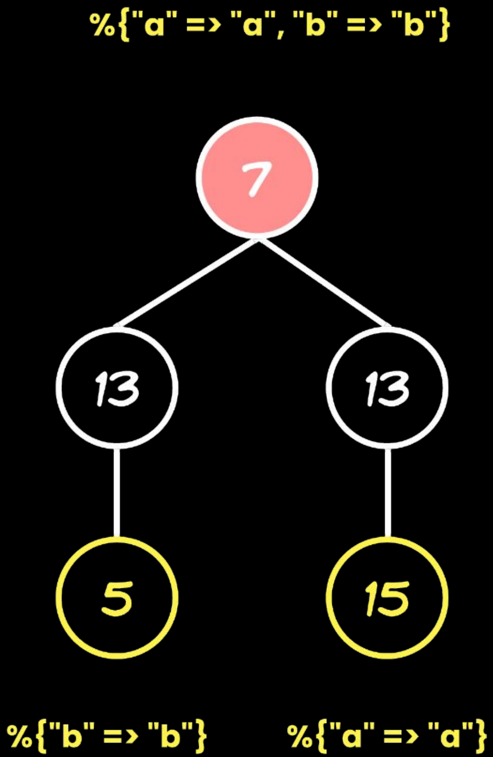
%{"a" => "a", "b" => "b"}

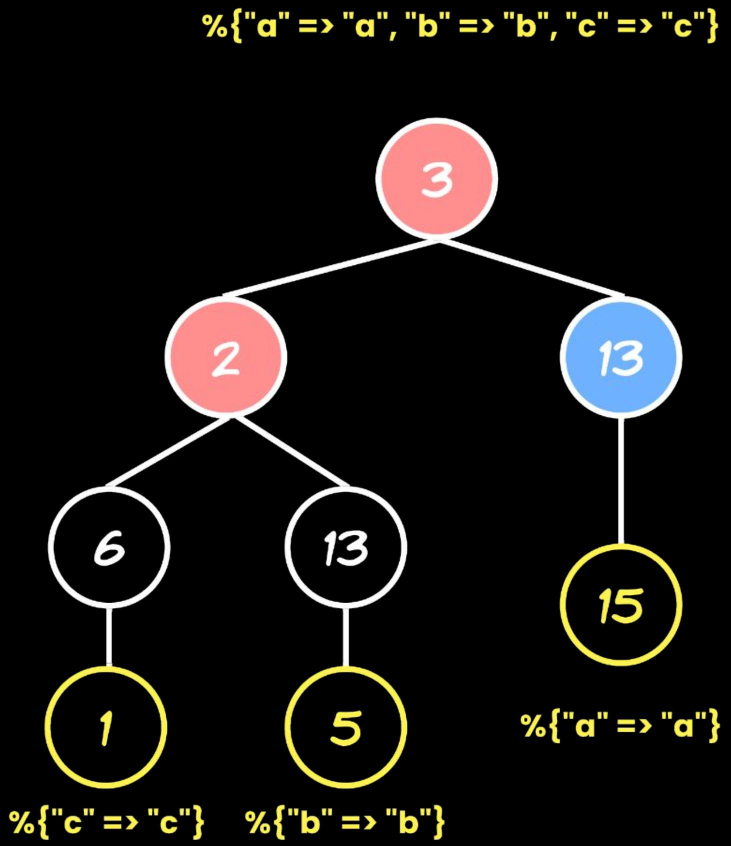


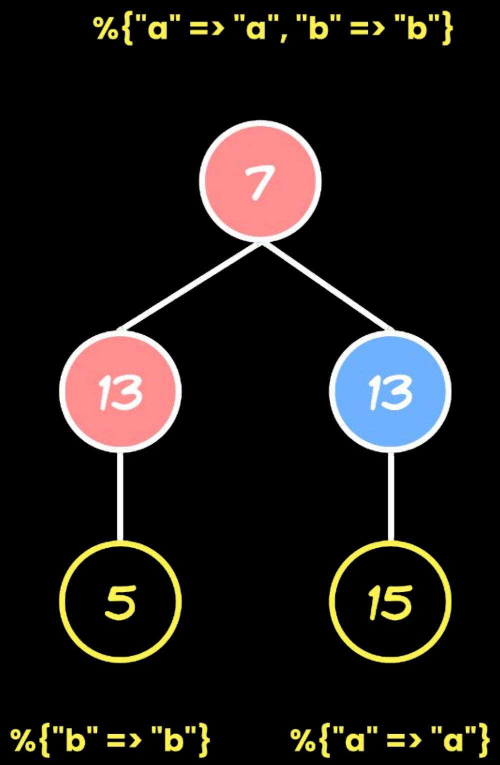

%{"a" => "a", "b" => "b", "c" => "c"}
3
2
13
6
13
15
1
5
%{"a" => "a"}
%{"c" => "c"}
%{"b" => "b"}

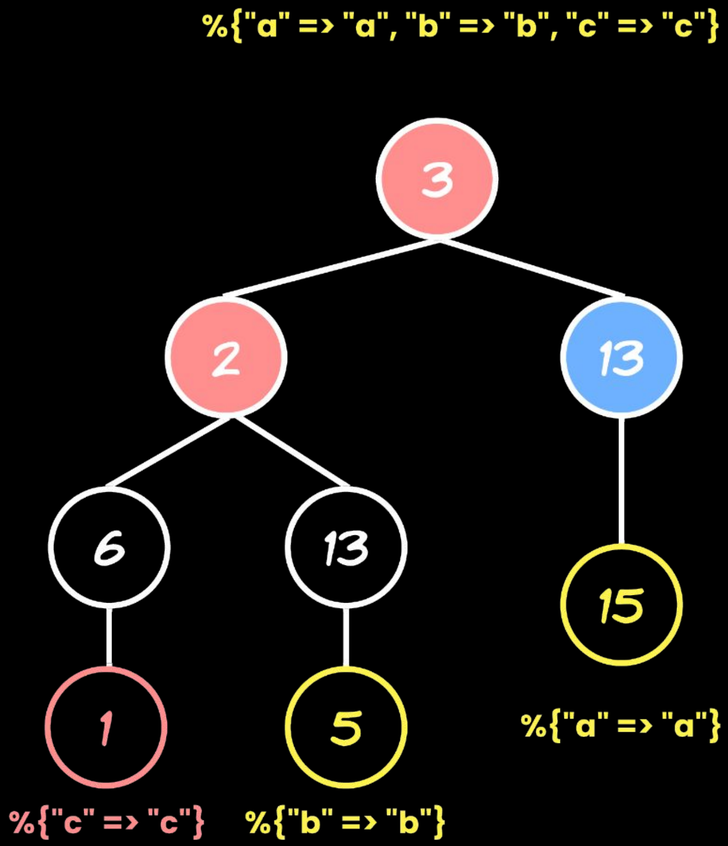
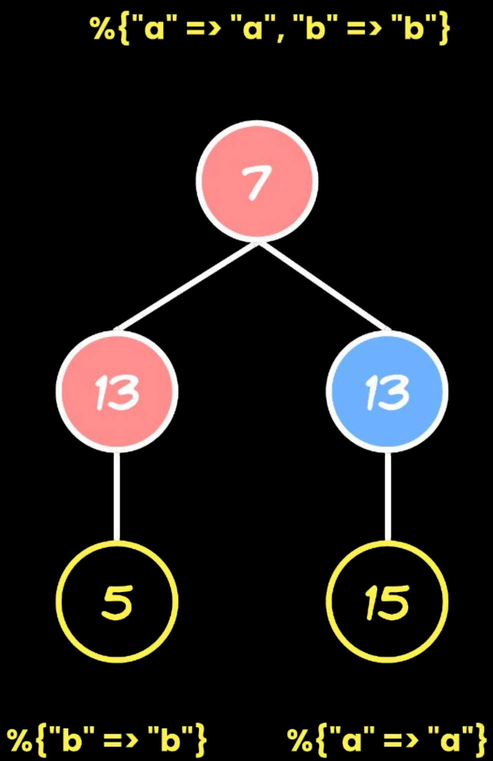

%{"a" => "a", "b" => "b"}
7
13
13
5
15
%{"b" => "b"}
%{"a" => "a"}

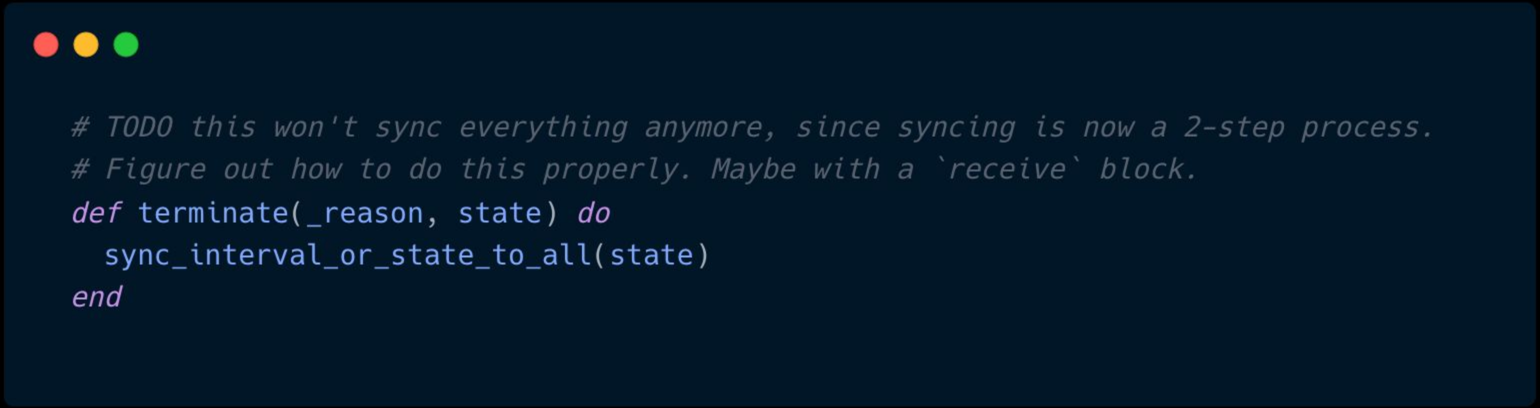

%{"a" => "a", "b" => "b", "c" => "c"}
3
2
13
6
13
15
1
5
%{"a" => "a"}
%{"c" => "c"}
%{"b" => "b"}


%{"a" => "a", "b" => "b"}
7
13
13
5
15
%{"b" => "b"}
%{"a" => "a"}


%{"a" => "a", "b" => "b", "c" => "c"}
3
2
13
6
13
15
1
5
%{"a" => "a"}
%{"c" => "c"}
%{"b" => "b"}

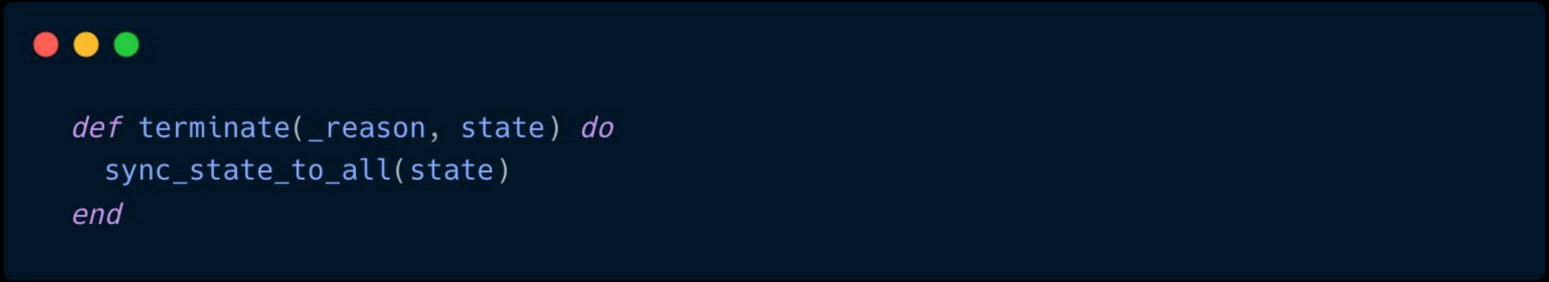

%{"a" => "a", "b" => "b"}
7
13
13
5
15
%{"b" => "b"}
%{"a" => "a"}


%{"a" => "a", "b" => "b", "c" => "c"}
3
2
13
6
13
15
1
5
%{"a" => "a"}
%{"c" => "c"}
%{"b" => "b"}

%{"a" => "a", "b" => "b"}
7
13
13
5
15
%{"b" => "b"}
%{"a" => "a"}



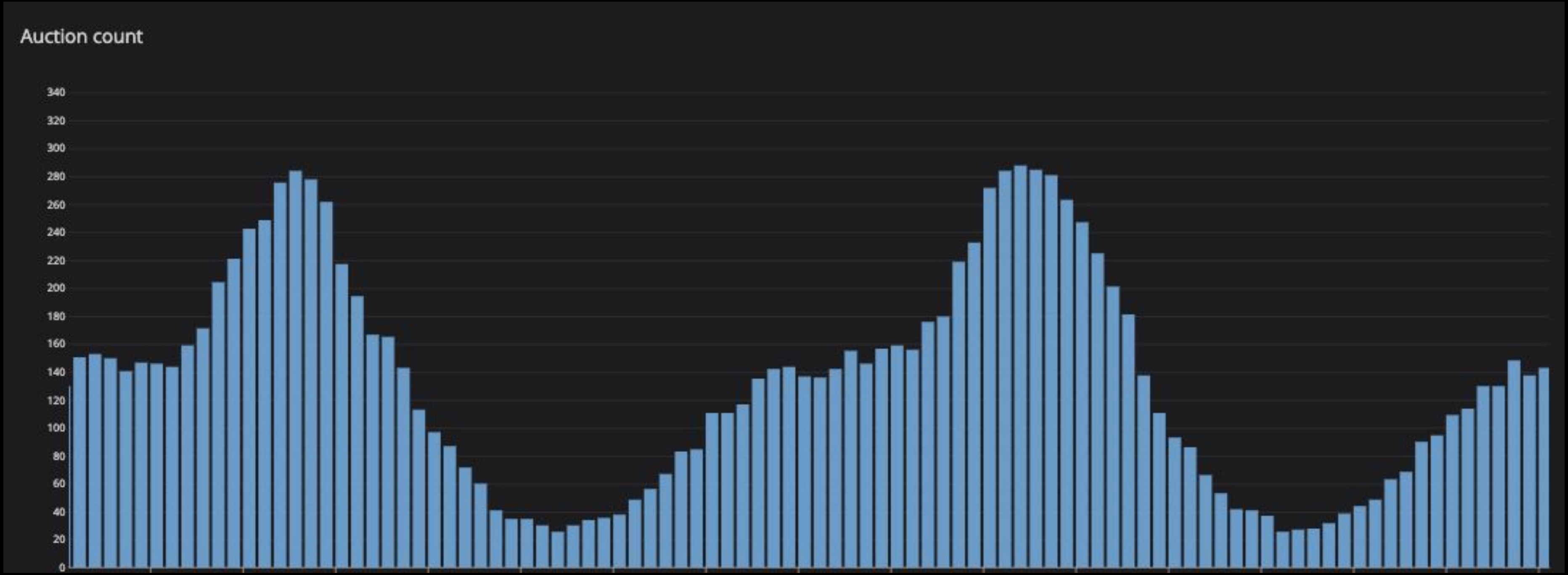
```elixir
# TODO this won't sync everything anymore, since syncing is now a 2-step process.
# Figure out how to do this properly. Maybe with a `receive` block.
def terminate(_reason, state) do
  sync_interval_or_state_to_all(state)
end
```

```elixir
def terminate(_reason, state) do
  sync_state_to_all(state)
end
```

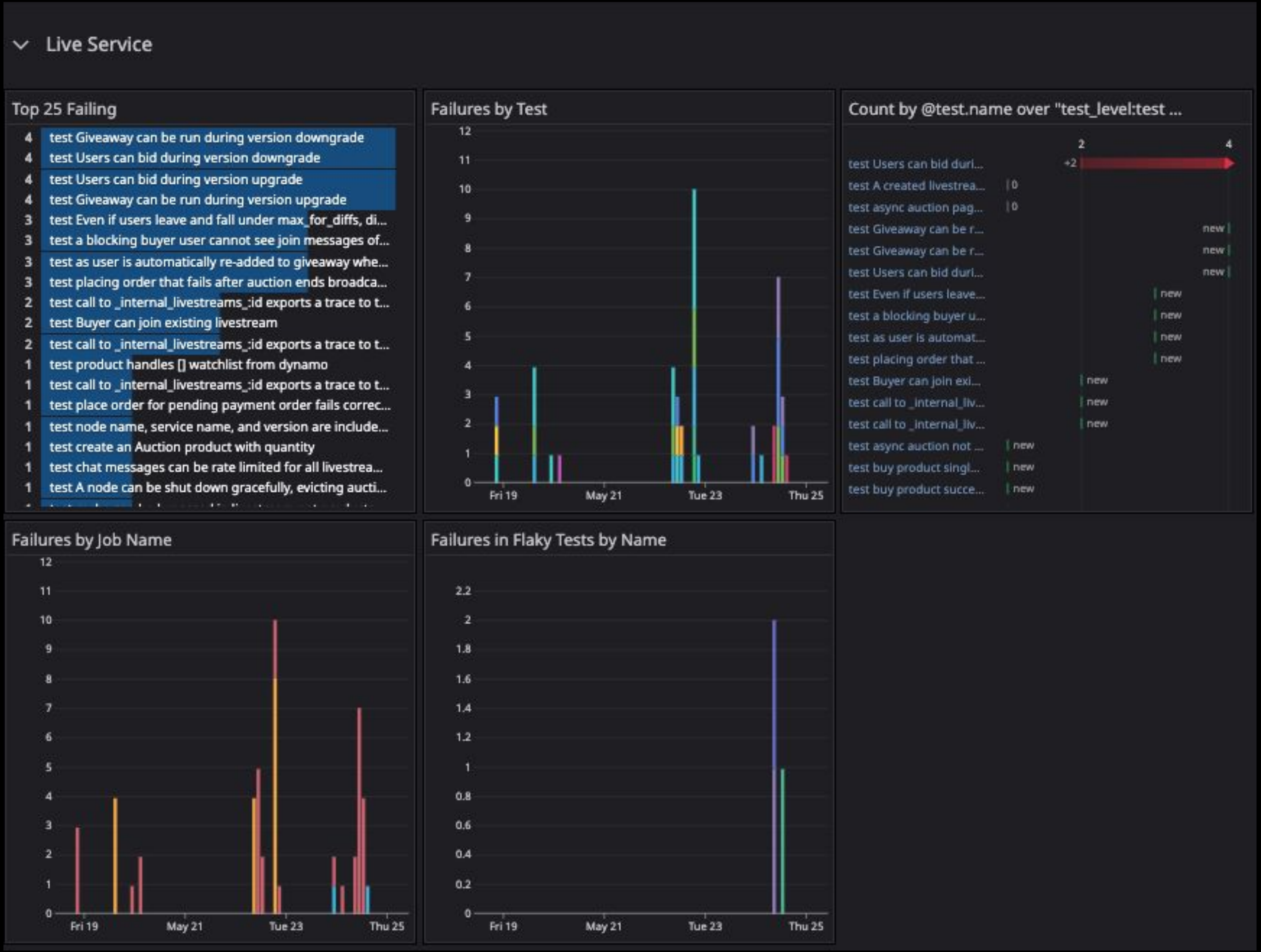


```elixir
def terminate(_reason, state) do
  sync_state_to_all(state)
end
```

https://github.com/Whatnot-Inc/delta_crdt_ex


Auction count
340
320
300
280
260
240
220
200
180
160
140
120
100
80
60
40
20
0

LET'S GO!

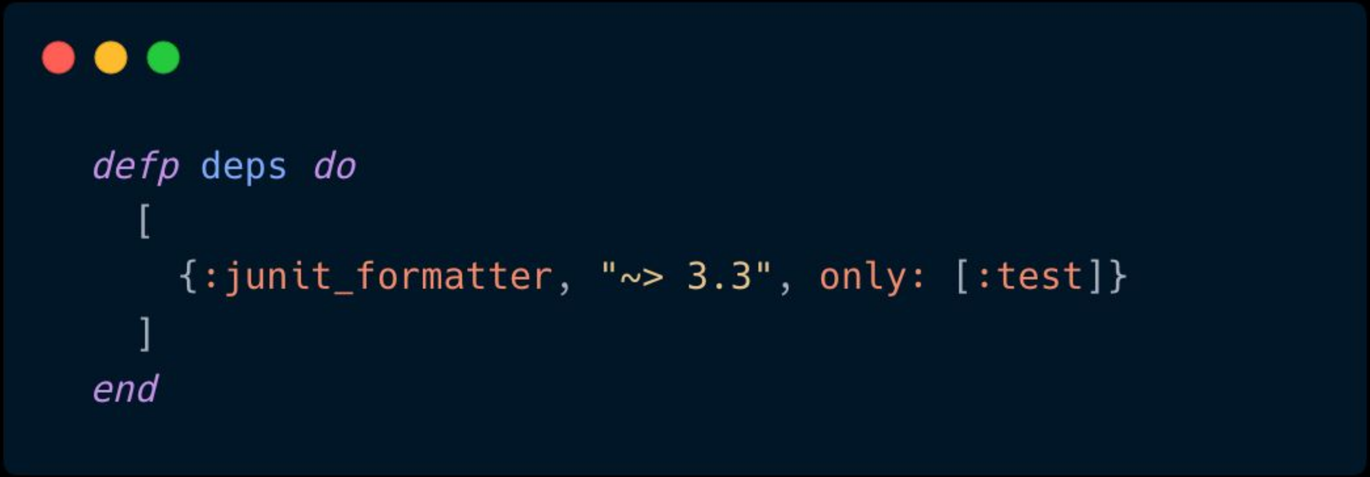


Top 25 Failing

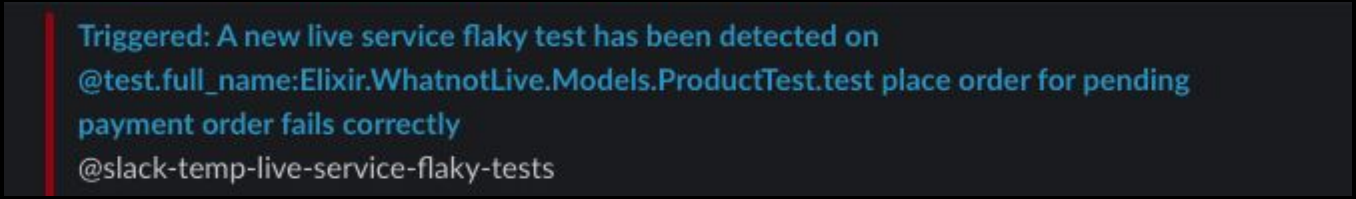
4	test Giveaway can be run during version downgrade
4	test Users can bid during version downgrade
4	test Users can bid during version upgrade
4	test Giveaway can be run during version upgrade
3	test Even if users leave and fall under max_for_diffs, di...
3	test a blocking buyer user cannot see join messages of...
3	test as user is automatically re-added to giveaway whe...
3	test placing order that fails after auction ends broadca...
2	test call to _internal_livestreams_:id exports a trace to t...
2	test Buyer can join existing livestream
2	test call to _internal_livestreams_:id exports a trace to t...
1	test product handles [] watchlist from dynamo
1	test call to _internal_livestreams_:id exports a trace to t...
1	test place order for pending payment order fails correc...
1	test node name, service name, and version are include...
1	test create an Auction product with quantity
1	test chat messages can be rate limited for all livestrea...
1	test A node can be shut down gracefully, evicting aucti...

Count by @test.name over "test_level:test ...

test Users can bid duri...	+2
test A created livestrea...	0
test async auction pag...	0
test Giveaway can be r...	new
test Giveaway can be r...	new
test Users can bid duri...	new
test Even if users leave...	new
test a blocking buyer u...	new
test as user is automat...	new
test placing order that ...	new
test Buyer can join exi...	new
test call to _internal_liv...	new
test call to _internal_liv...	new
test async auction not ...	new
test buy product singl...	new
test buy product succe...	new



```elixir
defp deps do
  [
    {:junit_formatter, "~> 3.3", only: [:test]}
  ]
end
```



Triggered: A new live service flaky test has been detected on @test.full_name:Elixir.WhatnotLive.Models.ProductTest.test place order for pending payment order fails correctly
@slack-temp-live-service-flaky-tests

Always Listen To Customers