

BROUGHT TO YOU BY AWS AND PARTNERS

GOTO **EDA DAY 2023**

Optimizing EDA Workflows: Real-time Serverless Communication

With Momento Topics

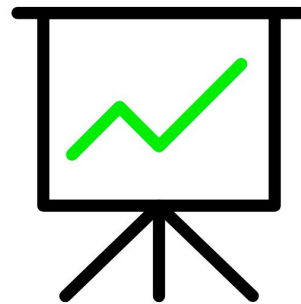




Table of contents

01.

About Momento

Who we are and what we do

02.

Definitions

Just to make sure we're on the same page

03.

Architecture

How I've built real-time notifications on AWS

04.

Architecture-less

New phrase of the day

05.

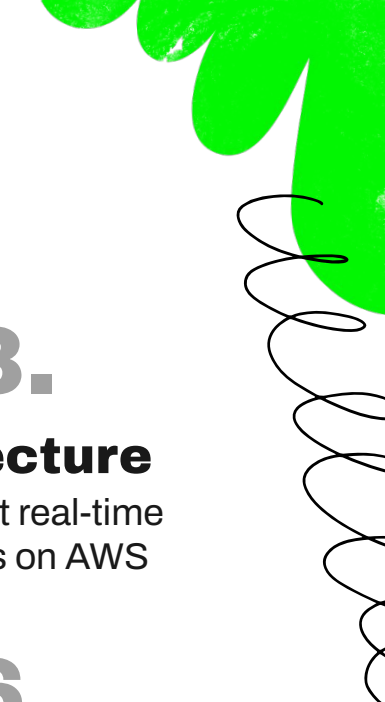
How it works

Ever wondered how to make a serverless service?

06.

Nifty Use Cases

Cool things I've built and cool things you can build



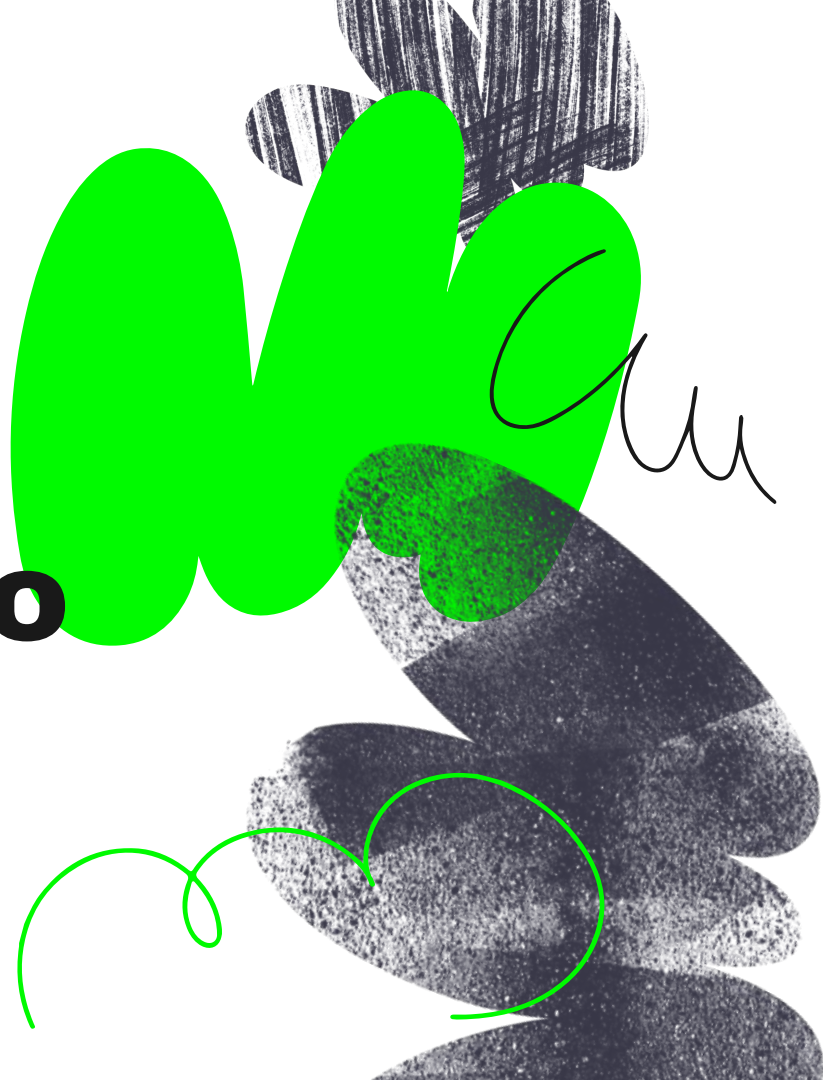


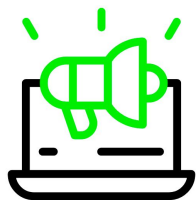
01.



About Memento

One of the cool guys



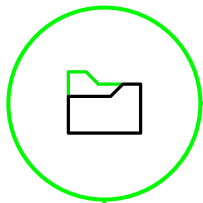


Our company

We believe in serverless. No, not in the market-y way people debate about on social media. But in the “we do the hard part so you can do the important part” kind of way. Auto-scaling, instance provisioning, always available, configuration. That’s on us.

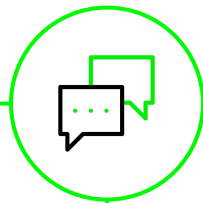


Our evolution



Caching

Our flagship product.
Fast, easy, legit.



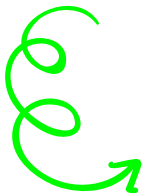
Topics

Event-messaging.
The reason we're here.



Vector Index

You do stuff with AI,
right? Right?



Guiding principles

Simple

Intuitive, easy to use
software



Security

Safe and secure
out-of-the-box



Serverless

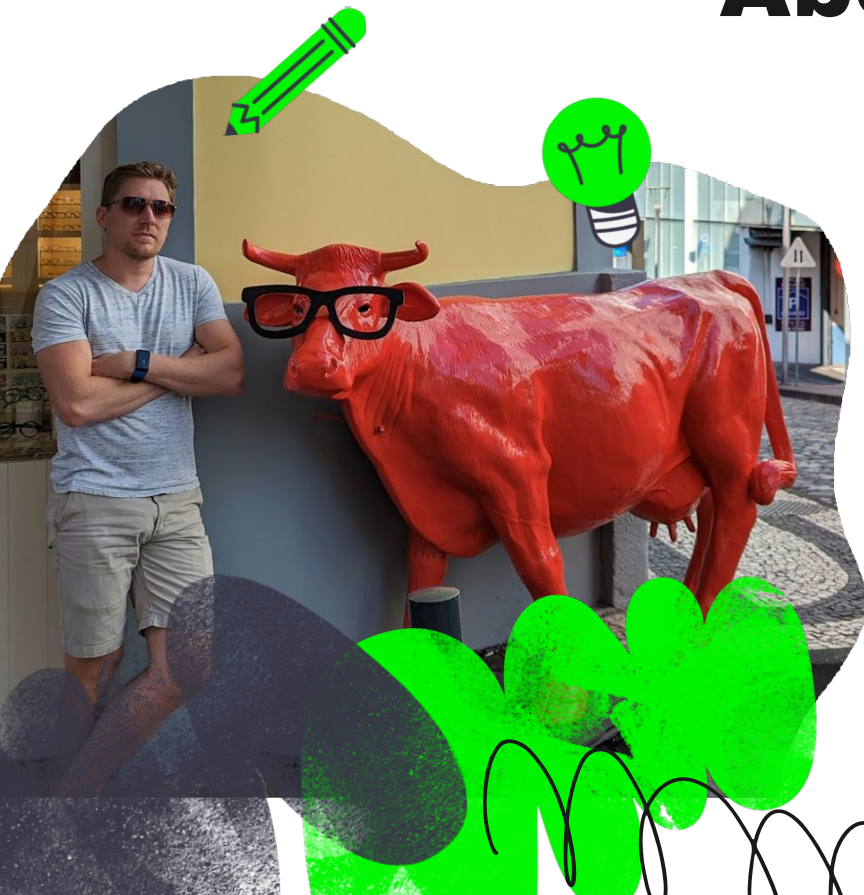
Don't worry about
scaling or configurations



Speed

Fast development,
faster performance

About Me

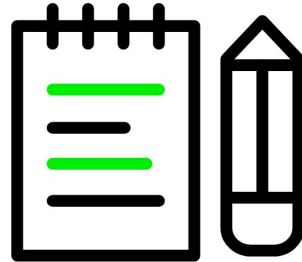


A few fun factoids

- Ecosystem Engineer
- AWS Serverless Hero
- I live on a farm
- Ready, Set, Cloud guy
- I do my own stunts

No, that is not a real cow in the picture.

Definitions



Real-time Notifications



Technical

- Event-triggered notifications
- Extremely low latency
- Two-way immediate communication

opinion alert!



ELI5

- Like when a doorbell rings
- And your dog barks right away
- Then the pizza delivery man yells “Pizza” and you yell “hold on, let me put away the dog”

WebSocket



Technical

- Full-duplex communication protocol
- Operates over single TCP connection
- Enables real-time data transfer
- Stateful
- Standardized by Internet Engineering Task Force (IETF)
- Uses the **ws** protocol



ELI5

- Two-way walkie talkie
- Uses one channel at a time
- Just like talking to your friends
- If your friend leaves, you know it
- Lots of smart people agreed this is the way to do it

Push Notification



Technical

- Server-initiated message
- Stateless and independent
- No memory of past interactions



ELI5

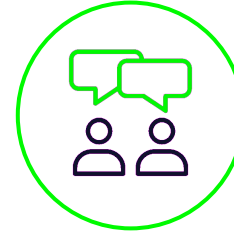
- Like a surprise letter from your friend
- Each letter comes on its own
- Every letter is a new conversation

Connections



Stateless

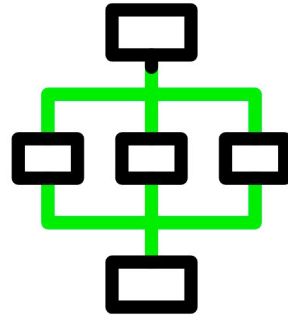
- No persistent connection
- No details about subscribers
- Each request is independent and requires full context



Stateful

- Constant open connection
- Can store/retrieve metadata on subscribers
- A third thing

Architecture



You have options

AWS AppSync

GraphQL subscriptions

AWS IoT Core

Publish and subscribe to devices

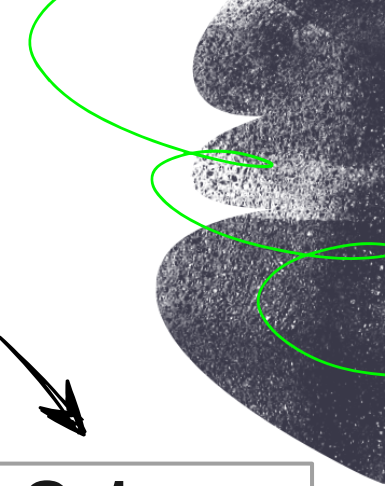
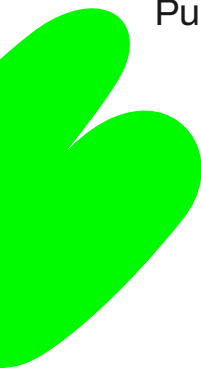
I've built this one before

AWS API Gateway

They have a WebSocket option

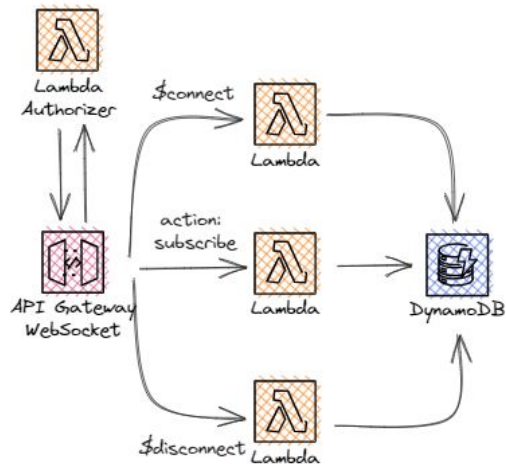
~~Amazon SNS~~

~~Fan-out pub sub~~

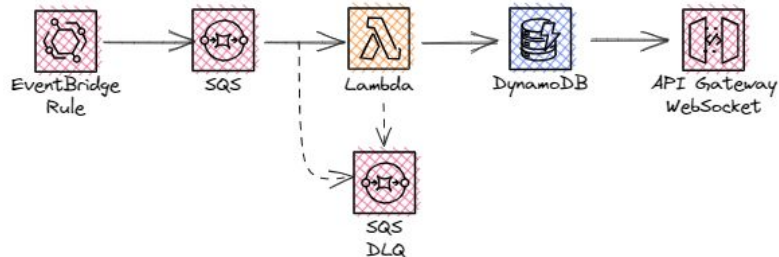


“Simple” Architecture

Connection Management



Posting to Connections

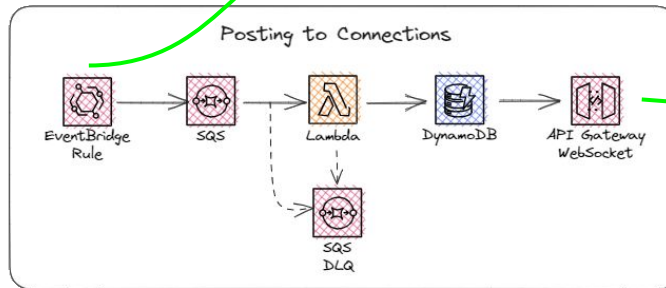
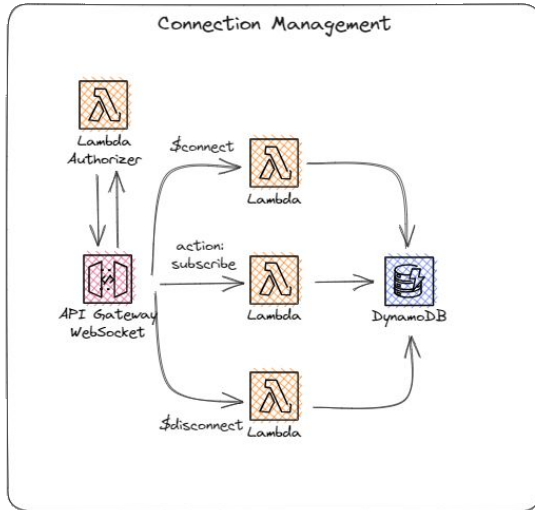
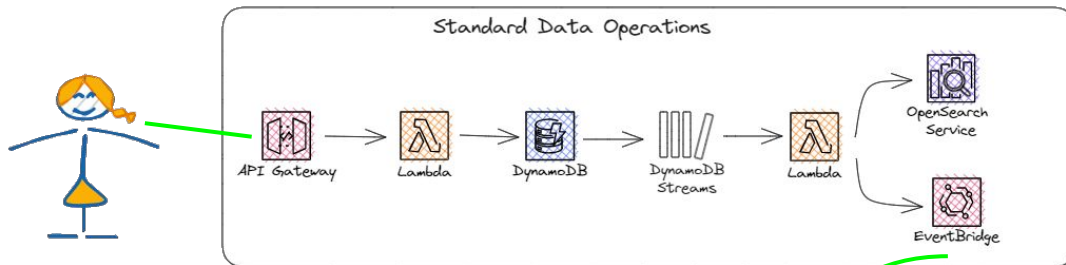


DynamoDB

Data Model

pk: { connection id } sk: "connection"
pk: { connection id } sk: "subscription" GSI: "chickens"
pk: { connection id } sk: "subscription" GSI: "goats"
pk: { connection id } sk: "subscription" GSI: "chickens"

Sending a message



Data Model

DynamoDB

- pk: { connection id } sk: "connection"
- pk: { connection id } sk: "subscription" GSI: "chickens"
- pk: { connection id } sk: "subscription" GSI: "goats"
- pk: { connection id } sk: "subscription" GSI: "chickens"

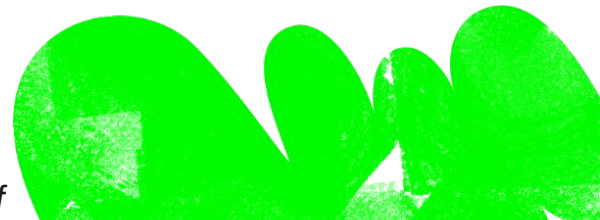
Source code



By the numbers

- 7 Lambda functions
- 593 lines of Infrastructure as Code
- 12 boxes on architecture diagrams
- 1 piece of functionality*

** kind of*



Architecture -less



Momento Topics

Bio

Age: 7 months (released in March)

SDKs: JS, Web, .NET, Python, Go, HTTP

Location: All regions

Cost: \$0.50/GB data transfer

Protocol: gRPC

Motivations



Qualifications

Scalability 90%



Speed 90%



Ease of Use 100%





Shift in focus



Publishers

Who is going to send
notifications?

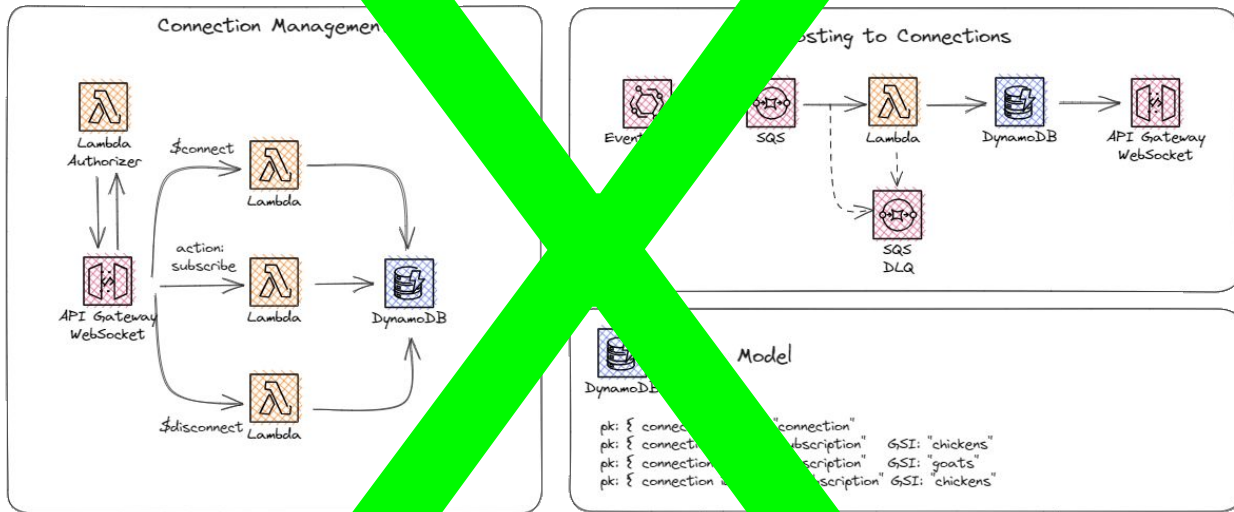
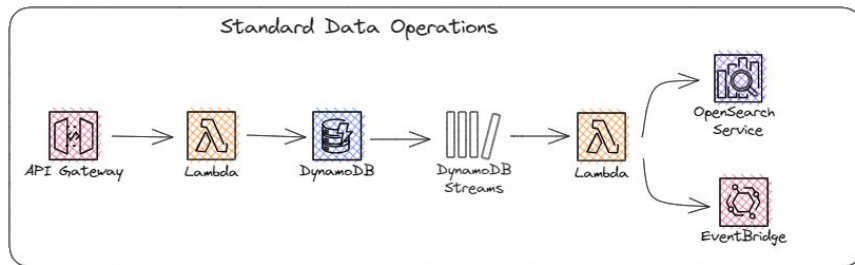


Subscribers

Who is going to receive
them?

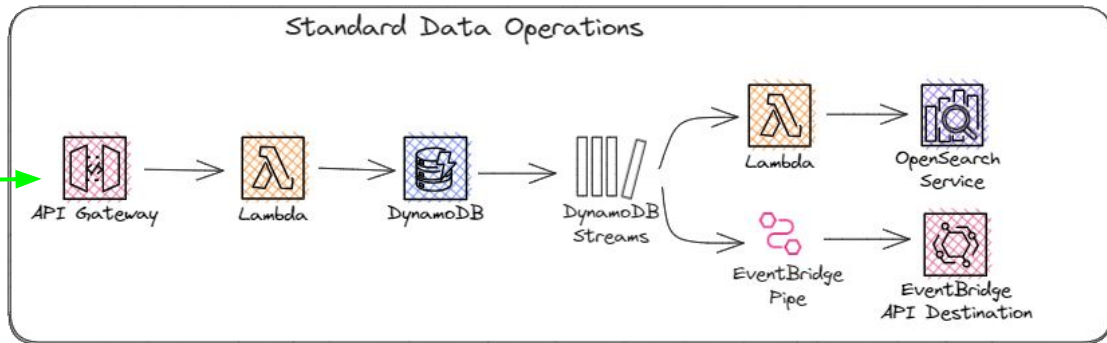


Sending a message



There is no architecture*

```
await topicClient.subscribe('farm-animals', 'goats', {  
  onItem: (msg) => myHandler(msg.valueString())  
});
```



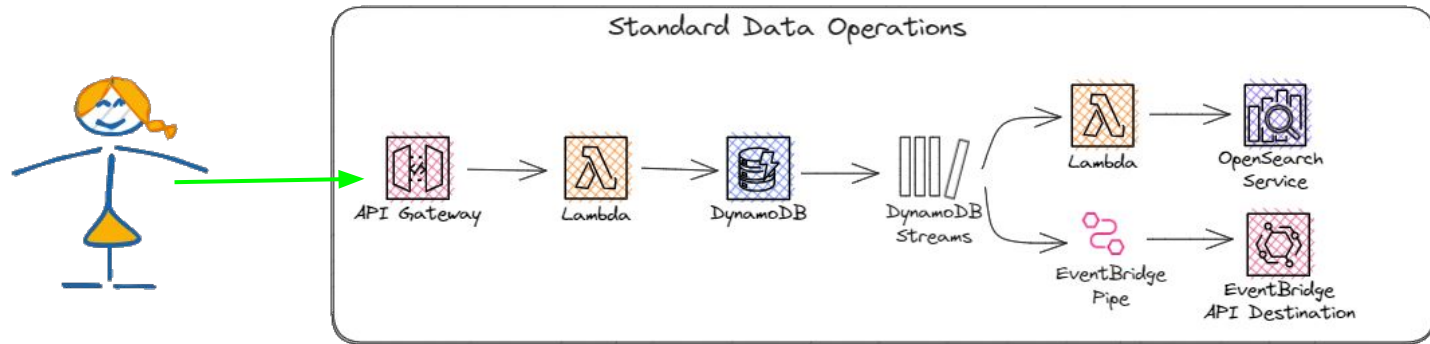
Momento
Topics



More on our
EventBridge API
Destinations

* that you care about

There is no architecture*

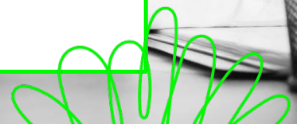


More on our
EventBridge API
Destinations

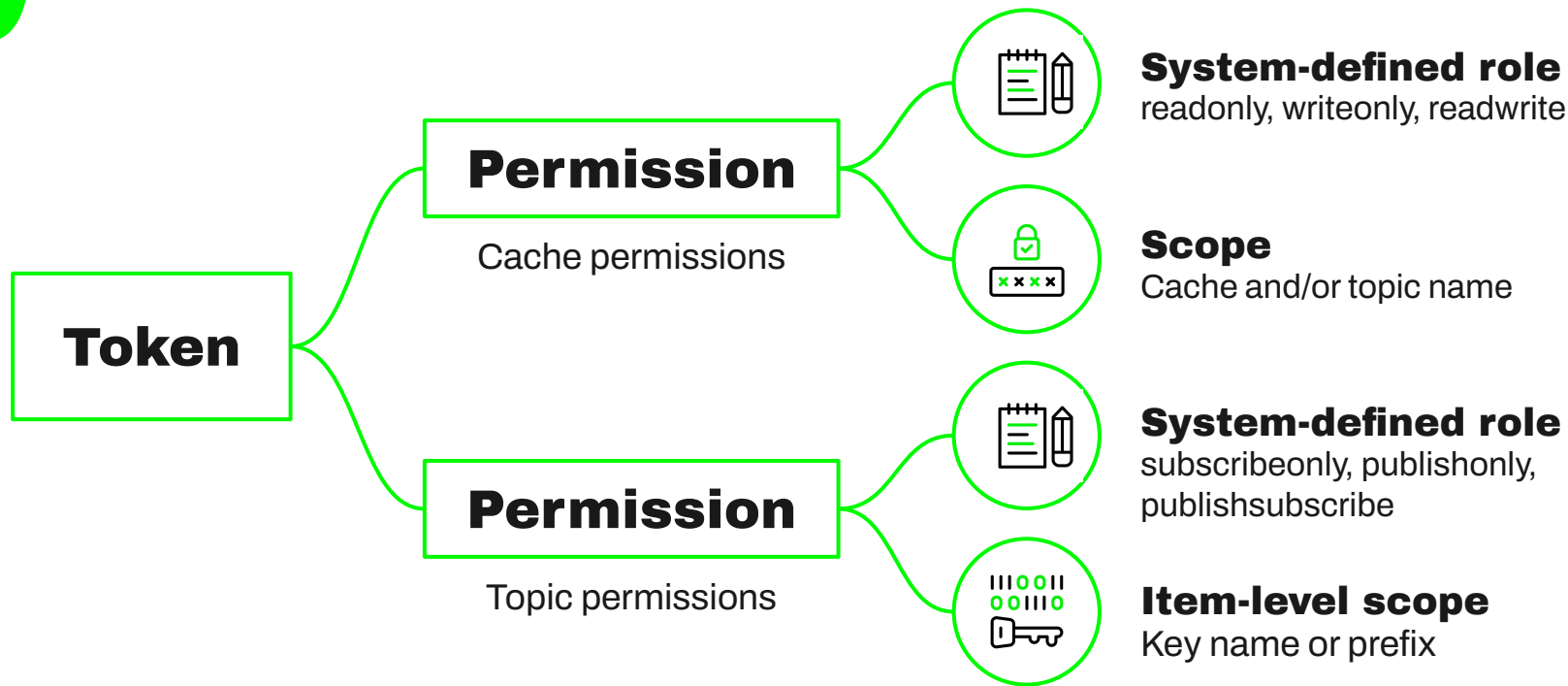
* that you care about



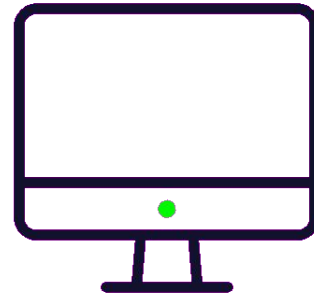
Where does security fit in?



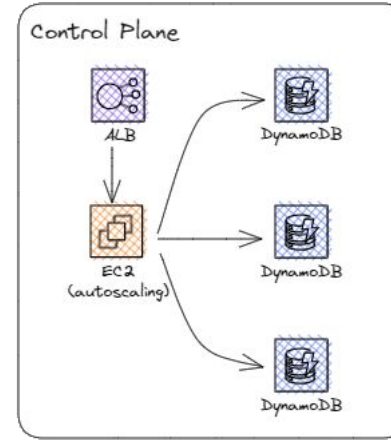
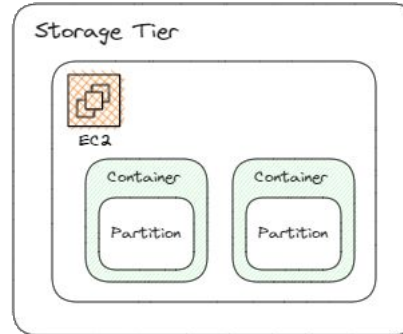
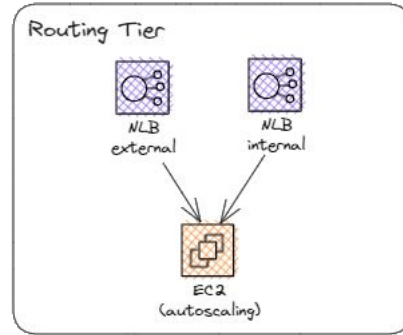
Fine-Grained Access Control



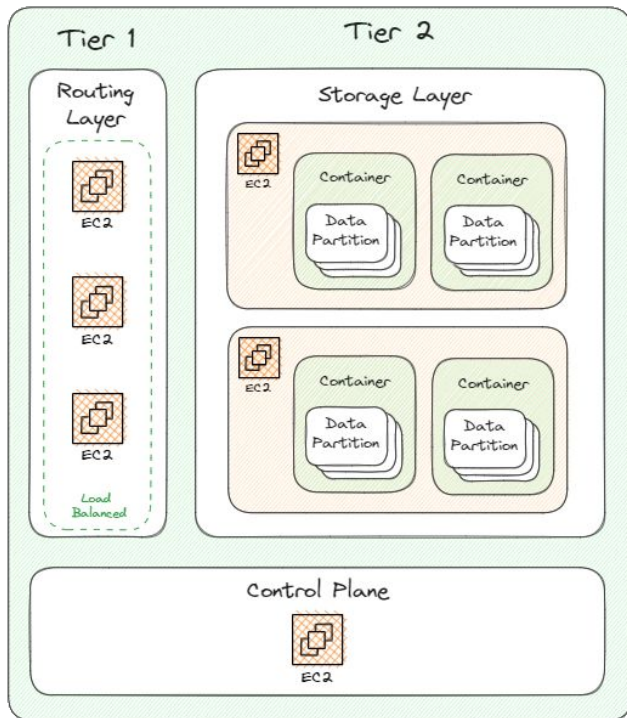
How it all works



The major components



Refocused components



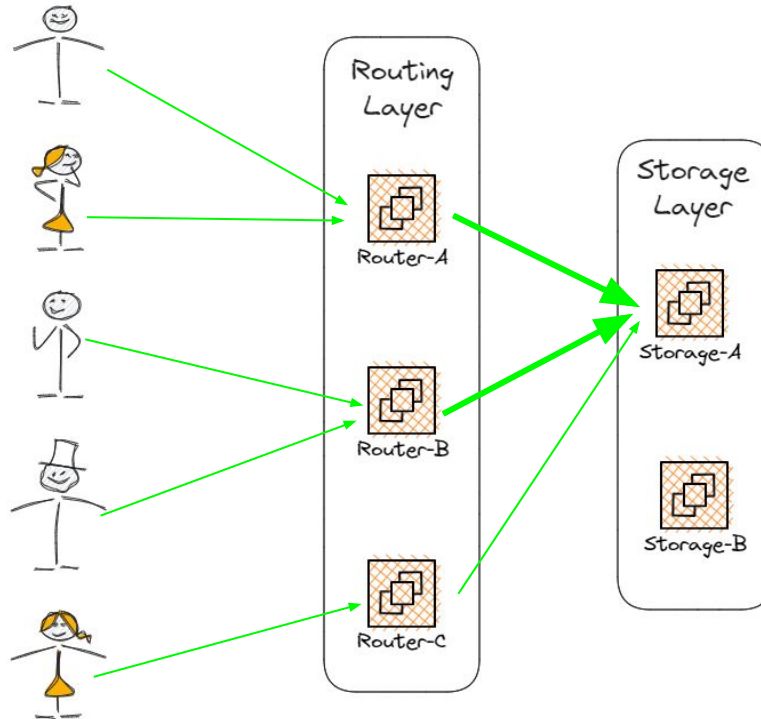
More on
rendezvous hashing



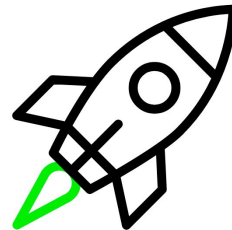
Example



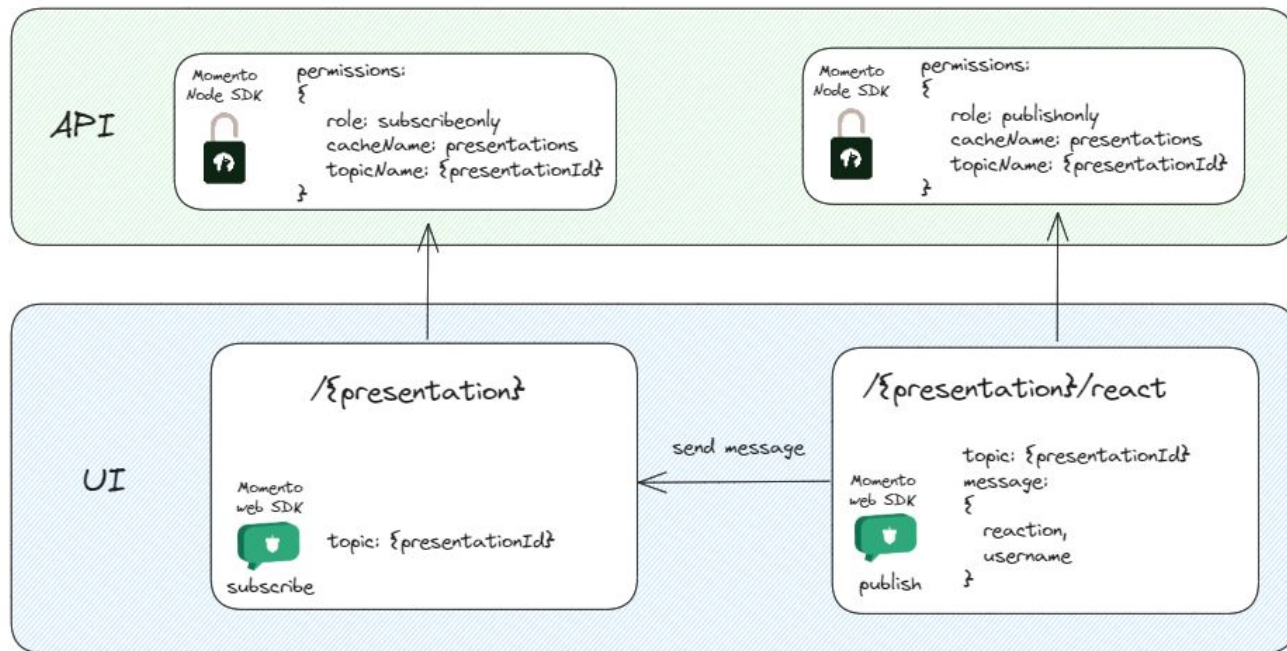
Tracking hail storm alerts for Dallas, TX



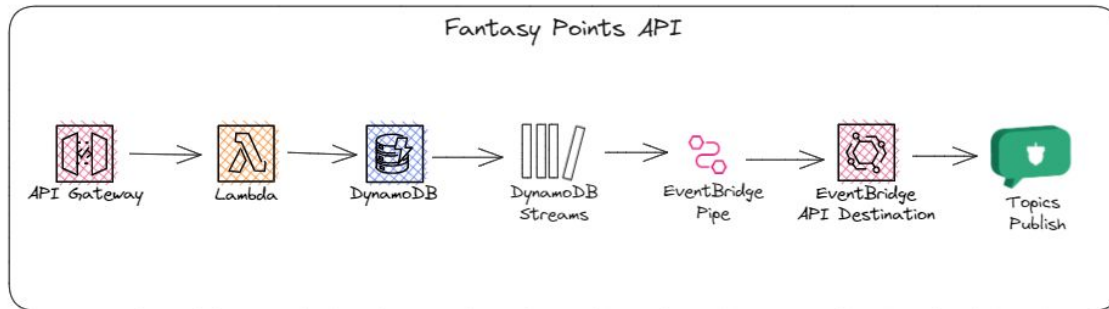
Cool things you can build



You're looking at it



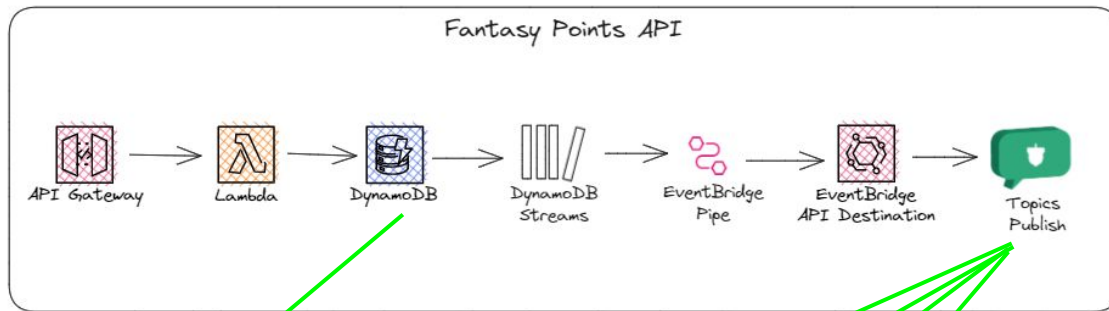
Fantasy Football App



```
for (const player of myTeam.players) {
  await topicClient.subscribe('american-football', player.id, {
    onItem: (msg) => updatePoints(player.id, Number(msg.valueString()))
  });
}
```



Fantasy Football App



POST /topics/american-football/kcc15
1.8

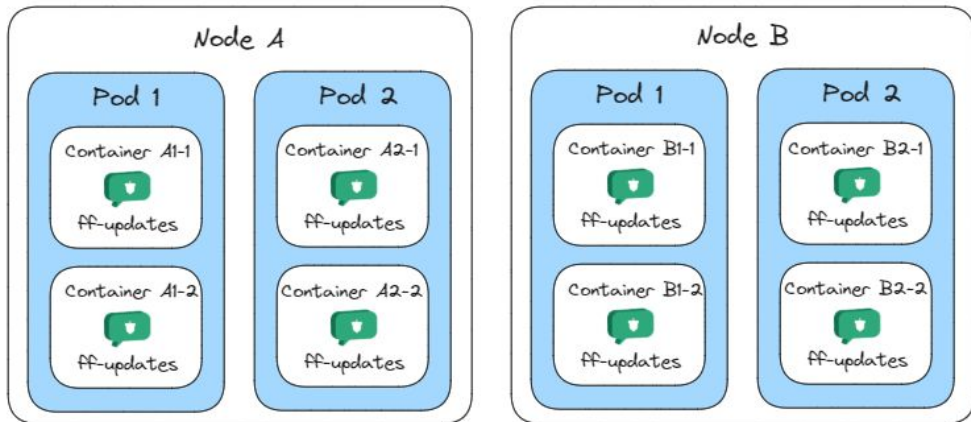
```
POST /games/{id}/stats
{
  "playerId": "KCC15",
  "type": "pass-yards",
  "value": "44"
}
```

```
{
  "pk": "KCC15",
  "sk": "stats#2023-10-01T20:55:04Z",
  "points": "1.8",
  "detail": {
    "type": "pass-yards",
    "value": "44"
  }
}
```

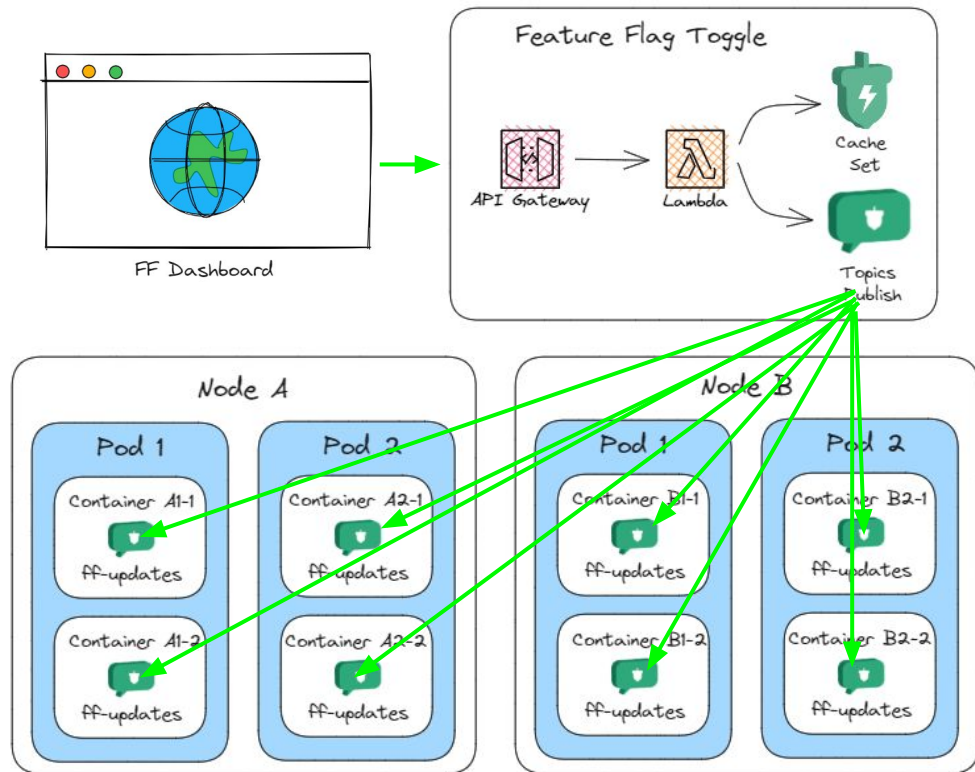


Feature Flag Manager

```
const flags = await cacheClient.dictionaryFetch('feature-flags', 'current');
await topicClient.subscribe('feature-flags', 'ff-updates', {
  onItem: (msg) => {
    const update = JSON.parse(msg.valueString);
    flags[update.flagName] = update.value;
  }
})
```



Feature Flag Manager



Chat



```
const chatHistory = await cacheClient.listFetch('chats', roomName);

const msg = JSON.stringify({ username, message });
cacheClient.listPushFront('chats', roomName, msg);
topicClient.publish('chats', roomName, msg);

});
```

Web App // Server Side



```
{
  permissions: [
    {
      role: readwrite,
      cacheName: chats,
      item: { key: <chat room name> }
    },
    {
      role: publishsubscribe,
      cacheName: chats,
      topicName: <chat room name>
    }
  ]
}
```

Momento instant messenger ⚡

iamdev: hey, I need help building chat into my app

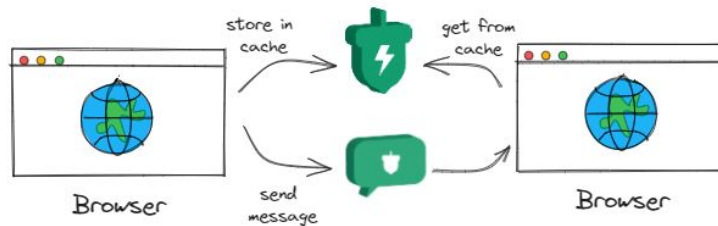
mo: I got you fam

You are logged in as iamdev

Type your message here

Send

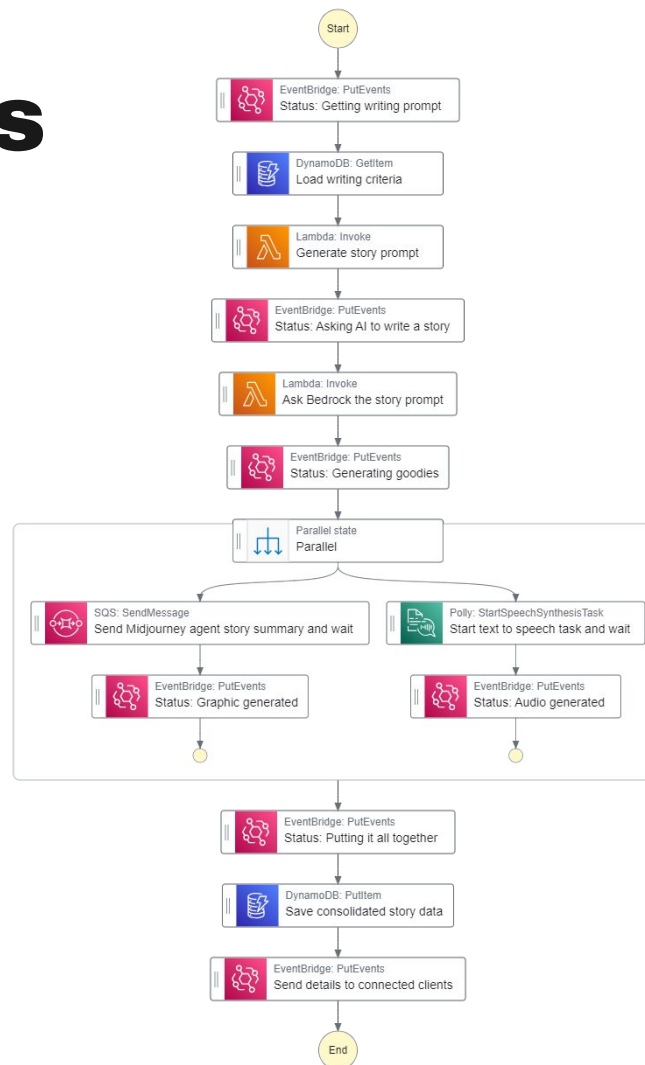
Web App // Client Side



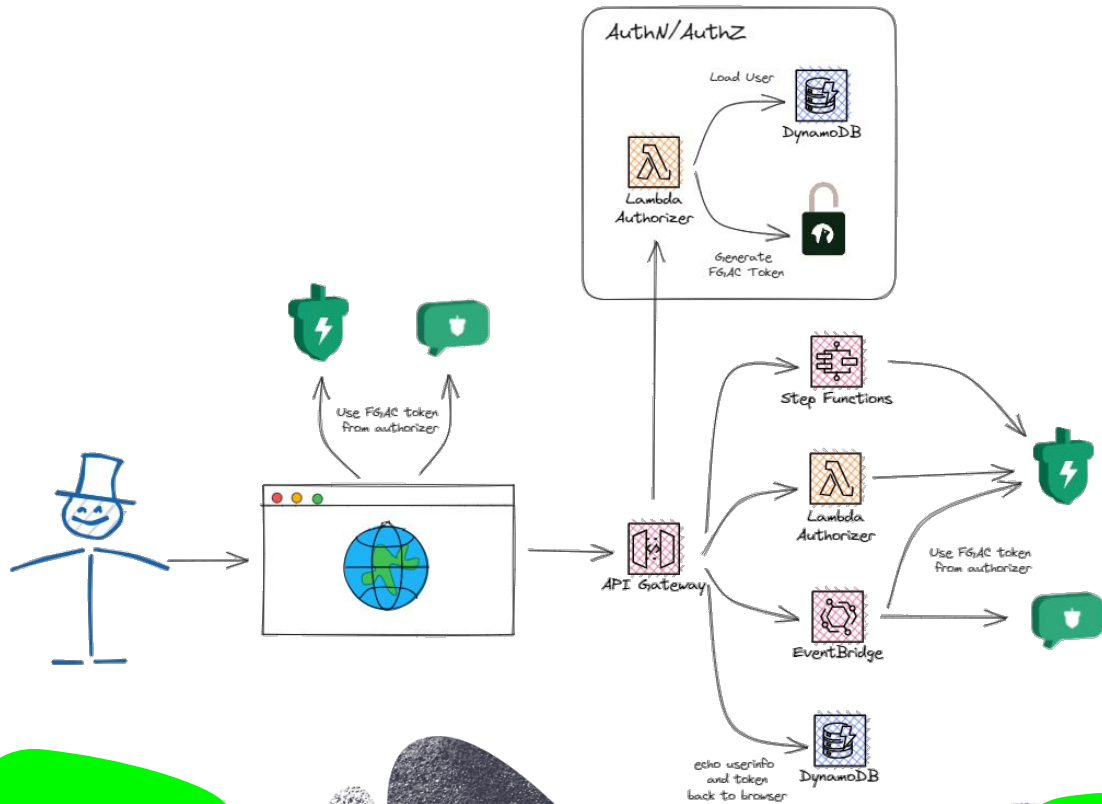
Status Updates



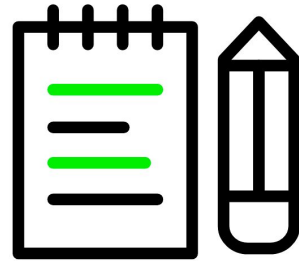
Status Updates



The cool thing about auth



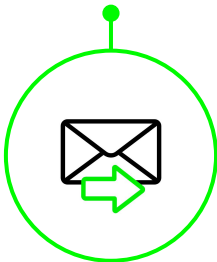
Super fast recap



What did we learn?

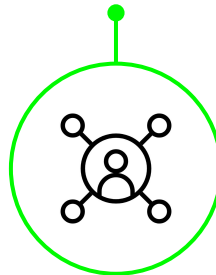
Real-time notifications

Not exclusive to WebSockets



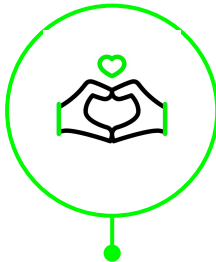
Crazy amount of use cases

Connects everything



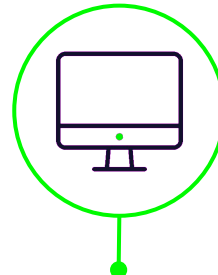
Topics

Pretty darn cool managed real-time notification service that removes boxes from your architecture diagrams



How serverless services work

Yes, there are servers in serverless



Thanks!

Do you have any questions?

@allenheltondev
gomomento.com



Learn more about Topics!



Don't forget to
vote for this session
in the **GOTO Guide app**