DYNA

Dyalog North America - 7 April 2025

The Many Faces of Jarvis

Brian Becker
APL Tools Architect
Dyalog, LTD

Agenda

- Jarvis overview
- Case studies and evolution with intermingled demos

What is Jarvis?

- Lets ask Anthropic's Claude

What is Jarvis?

Dyalog's Jarvis is a specialized library designed for building graphical user interfaces (GUIs) in the APL programming language, specifically for Dyalog APL. Let me explain what makes it particularly interesting and significant.

At its core, Jarvis provides a way to create web-based user interfaces using HTML5, CSS, and JavaScript, but with the unique ability to control everything from APL. This is quite remarkable because APL is known for its array-oriented programming and mathematical notation, not typically for building modern web interfaces.

Think of Jarvis as a bridge between two worlds: the powerful numerical and array processing capabilities of APL on one side, and modern web technologies on the other. It's named after Tony Jarvis, a significant figure in the APL community who contributed greatly to making APL more accessible and practical for real-world applications.

What is Jarvis?

Dyalog's Jarvis is a specialized library designed for building graphical user interfaces (GUIs) in the APL programming language, specifically for Dyalog APL. Let me explain what makes it particularly interesting and significant.

At its core, Jarvis provides a way to create web-based user interfaces using HTML5, CSS, and JavaScript, but with the unique ability to control everything from APL. This is quite remarkable because APL is known for its array-oriented programming and mathematical notation, not typically for building modern web interfaces.

Think of Jarvis as a bridge between two worlds: the powerful numerical and array processing capabilities of APL on one side, and modern web technologies on the other. It's named after Tony Jarvis, a significant figure in the APL community who contributed greatly to making APL more accessible and practical for real-world applications.

What is Jarvis?

Dyalog's Jarvis is a specialized library designed for building graphical user interfa UIs) in the APL programming language, specifically for Dyalog APL. Let me explai rticularly interesting and significant.

At its core, Jarvis provides a way to create we CSS, and JavaScript, but with the unique abili is quite remarkable because APL is known fo mathematical notation, not typically for building m

Thi worlds: the powerful numerical and array processing capa ide, and modern web technologies on the other. It's named after Tony Jarvis, nt figure in the APL community who contributed greatly to making APL more accessible and practical for real-world applications.

HALLUCINATIONS!

What is Jarvis?

- Lets ask Anthropic's Claude
- How about Google's Gemini?

Dyalog's Jarvis is a web service framework built in Dyalog APL. Here's a breakdown of its key features and purpose:

- **Web Service Framework:**
  - Jarvis enables Dyalog APL developers to easily create web services. This allows APL functions to be accessed via standard web protocols.
  - It supports both JSON and RESTful APIs, making it versatile for various web application needs.
- **Purpose:**
  - The primary goal of Jarvis is to simplify the process of exposing APL code to the web or local networks.
  - It bridges the gap between APL's powerful computational capabilities and the accessibility of web technologies.

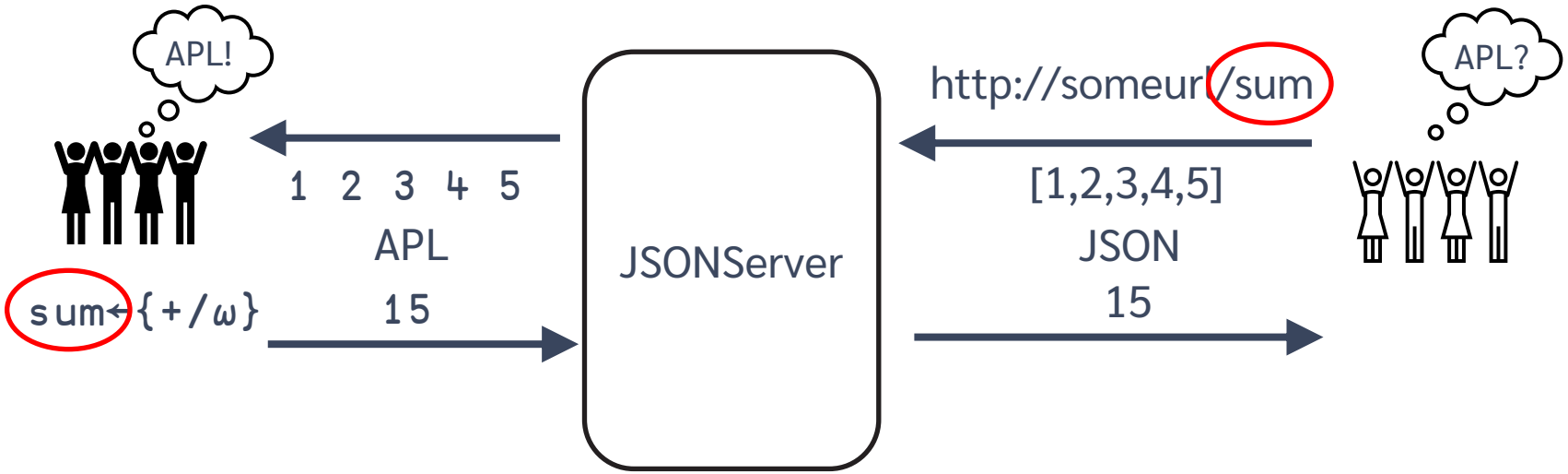
What is Jarvis?

- Lets ask Anthropic's Claude
- How about Google's Gemini?
- JSON and REST service (vis)
- APL-based web service framework supporting 2 paradigms
  - JSON – just like calling an APL function
  - REST – for the management of resources
- Nearly every new Dyalog APL project uses Jarvis

Web Service

A web service is a standardized way for applications to communicate and exchange data over a network, typically the internet. It allows different systems, built on various programming languages and platforms, to interact seamlessly.

It started as a simple concept



Clients

- Anything that can send and receive HTTP messages
  - Phone app
  - Browser/JavaScript
  - C#
  - Python
  - Even APL

Let the UI experts do the UI

- Jarvis doesn't do UI – it returns a data payload
- There are a bazillion frameworks to develop UI
- There are far more non-APL resources available to do UI
- Jarvis and EWC give you options on how much and what sort of UI you need to develop

Quick Demo

Then people started using it…

- Guess what? People are creative…
  - Can you make it do REST?
  - Can it serve HTML and other static content?
  - Can I use it to upload files?
  - Etc, etc, etc
- So, we added features and functionality

Take a REST

REST (Representational State Transfer) is an architectural style for designing networked applications. It uses a stateless, client-server communication protocol, typically HTTP, to interact with resources. RESTful APIs rely on standard HTTP methods, like:

- **GET**: Retrieve data.
- **POST**: Create new data.
- **PUT**: Update or create data.
- **DELETE**: Remove data.
- **PATCH**: Partially update data.

JSONServer becomes Jarvis

Adding REST support

- In REST mode, Jarvis manages "resources"
- Resource is specified in the request URL
  https://abc.com/customers/ID
- Action is specified by the HTTP method

Method	Action
GET	Read
POST	Create or Update
PUT	Replace
PATCH	Partial update
DELETE	Remove

JSON vs REST

JSON

Endpoints are APL functions

```
∇ r←GetCustomer ID
```

REST

Endpoints are resources

`https:/abc.com/customers/ID`

Write a function for each HTTP method you want to support

The function parses the request to determine the resource and acts appropriate for the HTTP method

To REST or not to REST?

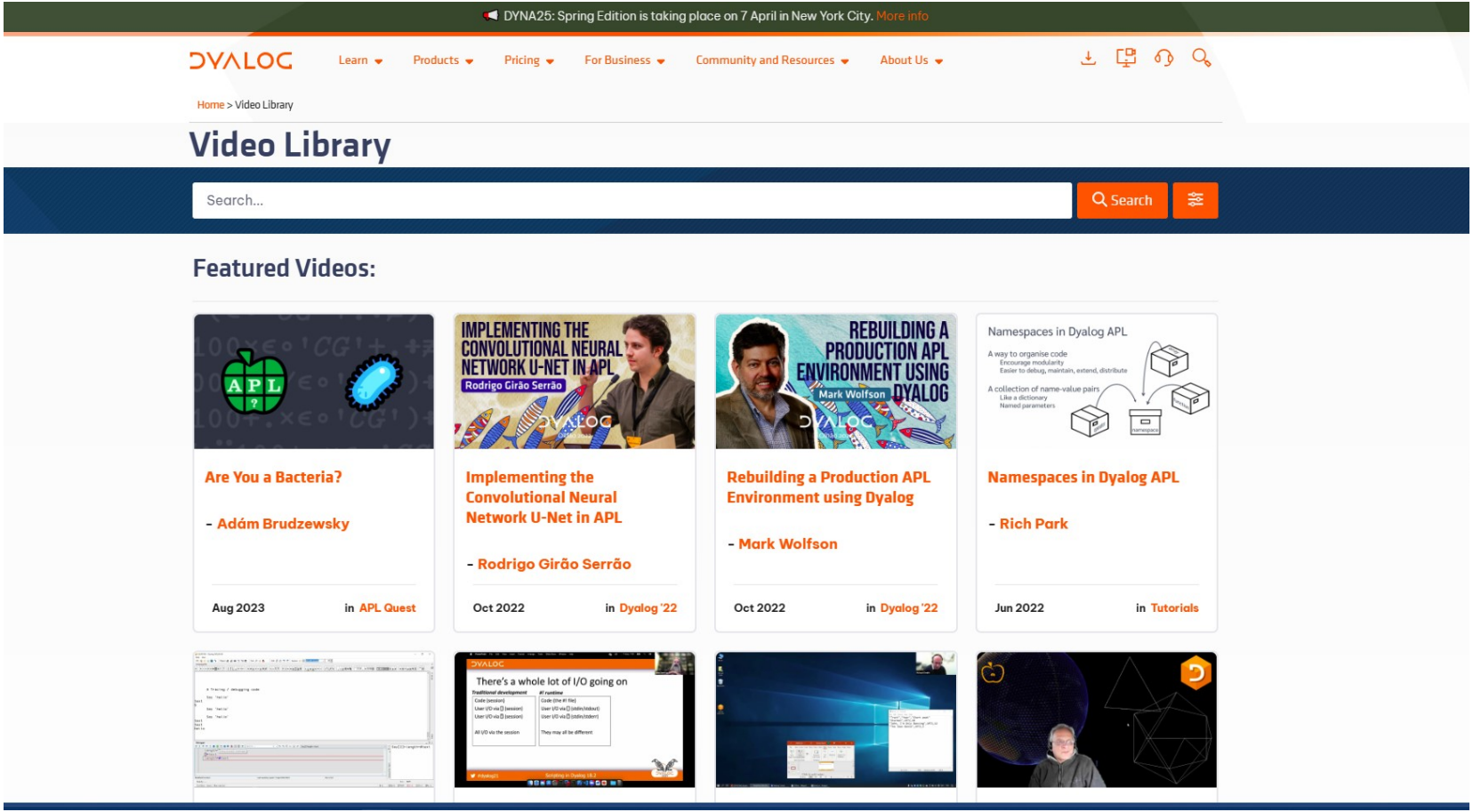
Many web service APIs (application programming interfaces) use REST

- GitHub, Google, many LLMs (OpenAI, Anthropic, Google Gemini)
- Designing a REST API takes some thought to get it right

DCMS

- DCMS is Dyalog's content management system that supplies data for the Dyalog website and video library
- Rich Park is using Jarvis' REST paradigm to provide an API to manage the data

```
[
  {
    "category": "",
    "description": "Write a dfn to produce a vector of the first n odd numbers.\n\nWiki: https://apl.wiki/APL_Quest\nCode:
https://github.com/abrudz/apl_quest/blob/main/2013/1.apl\nChat: https://chat.stackexchange.com/transcript/52405?m=60343161#60343161\nQuest:
https://problems.tryapl.org/psets/2013.html?goto=P1_Seems_a_Bit_Odd_To_Me",
    "event": "APL Quest",
    "event_shortname": "apl-quest",
    "presented_at": "2022-02-01 00:00:00",
    "presenter": "Adám Brudzewsky",
    "published_at": "2022-02-08 00:00:00",
    "thumbnail": "https://i.ytimg.com/vi/Mj4wyLKrBho/maxresdefault.jpg",
    "title": "Seems a Bit Odd To Me",
    "url": "/video-library/watch?v=Mj4wyLKrBho",
    "youtube_id": "Mj4wyLKrBho"
  },
  {
    "category": "",
    "description": "Write a dfn which solves a set of linear equations. The left argument is a vector of the values for the equations and the right
argument is a matrix of the coefficients.\n\nWiki: https://apl.wiki/APL_Quest\nCode: https://github.com/abrudz/apl_quest/blob/main/2013/10.apl\nChat:
https://chat.stackexchange.com/transcript/52405?m=60845175#60845175\nQuest: https://problems.tryapl.org/psets/2013.html?goto=P10_Solution_Salvation",
    "event": "APL Quest",
    "event_shortname": "apl-quest",
    "presented_at": "2022-04-01 00:00:00",
    "presenter": "Adám Brudzewsky, Adám Brudzewsky",
    "published_at": "2022-04-13 00:00:00",
    "thumbnail": "https://i.ytimg.com/vi/w-rzx2VNqbY/maxresdefault.jpg",
    "title": "Solution Salvation",
    "url": "/video-library/watch?v=w-rzx2VNqbY",
    "youtube_id": "w-rzx2VNqbY"
  },
  {
    "category": "",
```

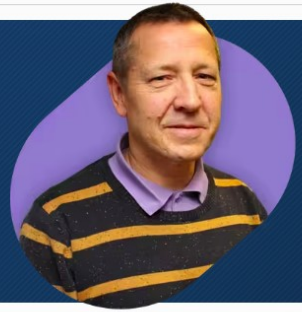

DYNA25: Spring Edition is taking place on 7 April in New York City. More info
DYALOG
Learn
Products
Pricing
For Business
Community and Resources
About Us
Home > Video Library
Video Library
Search...
Search
Featured Videos:
Are You a Bacteria?
- Adám Brudzewsky
Aug 2023
in APL Quest
IMPLEMENTING THE CONVOLUTIONAL NEURAL NETWORK U-NET IN APL
Rodrigo Girão Serrão
Implementing the Convolutional Neural Network U-Net in APL
- Rodrigo Girão Serrão
Oct 2022
in Dyalog '22
REBUILDING A PRODUCTION APL ENVIRONMENT USING DYALOG
Mark Wolfson
Rebuilding a Production APL Environment using Dyalog
- Mark Wolfson
Oct 2022
in Dyalog '22
Namespaces in Dyalog APL
Namespaces in Dyalog APL
- Rich Park
Jun 2022
in Tutorials
There's a whole lot of I/O going on

Home > About > Team Dyalog > Morten Kromberg

Morten Kromberg

Technical Director (CTO)

Joined Dyalog Ltd in April 2005. Based in Denmark.



About Morten | Blog Posts | Videos

The Road Ahead

In accordance with tradition, Morten looks briefly back over his shoulder before turning his gaze to the future, presenting his view of the road that lies before Dyalog and users of Dyalog APL. 00:00 Introduction 00:16 Celebration of 40 years of Dya...View

Using Packages

Following his presentation on Projects and Packages at Dyalog '22, Morten demonstrates a version of the Cider project management system that simultaneously supports two package managers – Tatin for packages implemented in Dyalog, and _NuGet_ for .N...View

The Road Ahead

How might Dyalog evolve in the years to come, and which technologies should you be trying to keep an eye on yourself? Morten attempts to answer these questions, and sets the scene for the technical presentations being given at Dyalog '22. 00:00 Over...View

```
res←Get req;allowed;nl
:If GLOBAL.debug   ⍝ :If here because prefer not to compute ⍕⎕TS for every request
    ⎕←'> GET REQUEST at ',⍕⎕TS
    ⎕←req.Endpoint
:EndIf
:Trap GLOBAL.debug↓0
    res←req.Response
    res.Headers←0 2⍴''
    res.Headers,←'Access-Control-Allow-Origin'service.AcceptFrom
    res.Headers,←'Access-Control-Allow-Headers' '*'

    req.Endpoint←{⍵≠⍨1(↑∨⌽)'/'≠⍵}req.Endpoint   ⍝ Remove multiple slashes
    :If req.Endpoint(⊃⍷⍨)'/videos'
        res←read.videos.Handle req
    :ElseIf 1=+/req.Endpoint∘≡¨'/person' '/organisation' '/event' '/event_type' '/presentation' '/presentation_type
        res←read.Table 1↓req.Endpoint
    :ElseIf 1=+/req.Endpoint∘≡¨'/presenters' '/dtv_events'
        res←CACHE⍎1↓req.Endpoint
    :ElseIf req.Endpoint≡'/version'
        res←Version
    :ElseIf req.Endpoint≡'/teapot'
        req.Fail 418
    :Else
        req.Fail 404
    :EndIf
:Else
    'Internal Server Error'req.Fail 500
:EndTrap
```

Portfolio and Risk Management

- A customer in Denmark uses Jarvis in JSON mode
  - It has a user interface developed in React (a popular JavaScript library)
  - It also exposes endpoints called directly (application to application)
- Added new `PostProcessFn` "hook" function

"Hook" functions

- Functions the user can specify to inject behavior at specific points in Jarvis' flow
  - AppCloseFn – when Jarvis is ending
  - AppInitFn – before Jarvis starts
  - AuthenticateFn – performs authentication on every request
  - PostProcessFn – after your endpoint has run, but before Jarvis responds
  - SessionInitFn – initializes a session, if using sessions
  - ValidateRequestFn – called on every request before any other processing

Quick Demo

qWC and HTMLInterface

- qWC is a product developed by Michael Hughes
- Michael needed to be able to serve up "static" content like CSS
- `HTMLInterface` setting originally only controlled whether Jarvis' internal demo page was active
- You can now specify a file or folder that contains HTML, CSS, JavaScript, image files, etc.

TryAPL

- Originally implemented in 2012 using MiServer
- Rewritten to use Jarvis in 2021

```
{
  "CodeLocation": "/app/TAE.apln",
  "IncludeFns":"Exec",
  "HTMLInterface":"/app/index.html"
}
```

ACBL

Jay Whipple is very involved with the ACBL (American Contract Bridge League) and uses Jarvis for 3 applications:

- Shark – implements the integration of game management and a remote third party game engine
- BridgeWar – a third party needed privileged access to ACBL player information. API took 30 minutes to set up
- ACBL Results Gateway – ACBL uploads hundreds of game files daily which are then validated and game results returned.

BIG

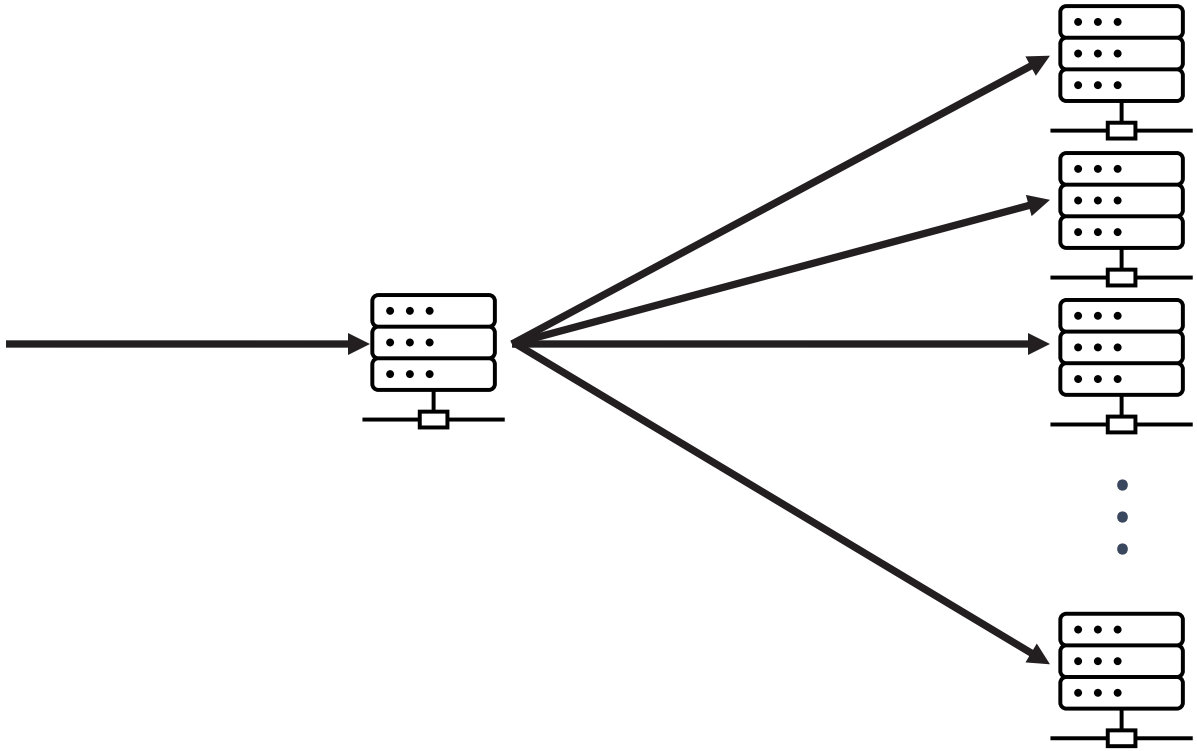
- Mark and Dexter used Jarvis to speed development of new dashboard capabilities, cutting development time to a fraction of what was able to be done in C#.

CITA

- Continuous Integration Testing in APL
- Uses Jarvis for its dashboard functions

Multi-tiered Jarvis

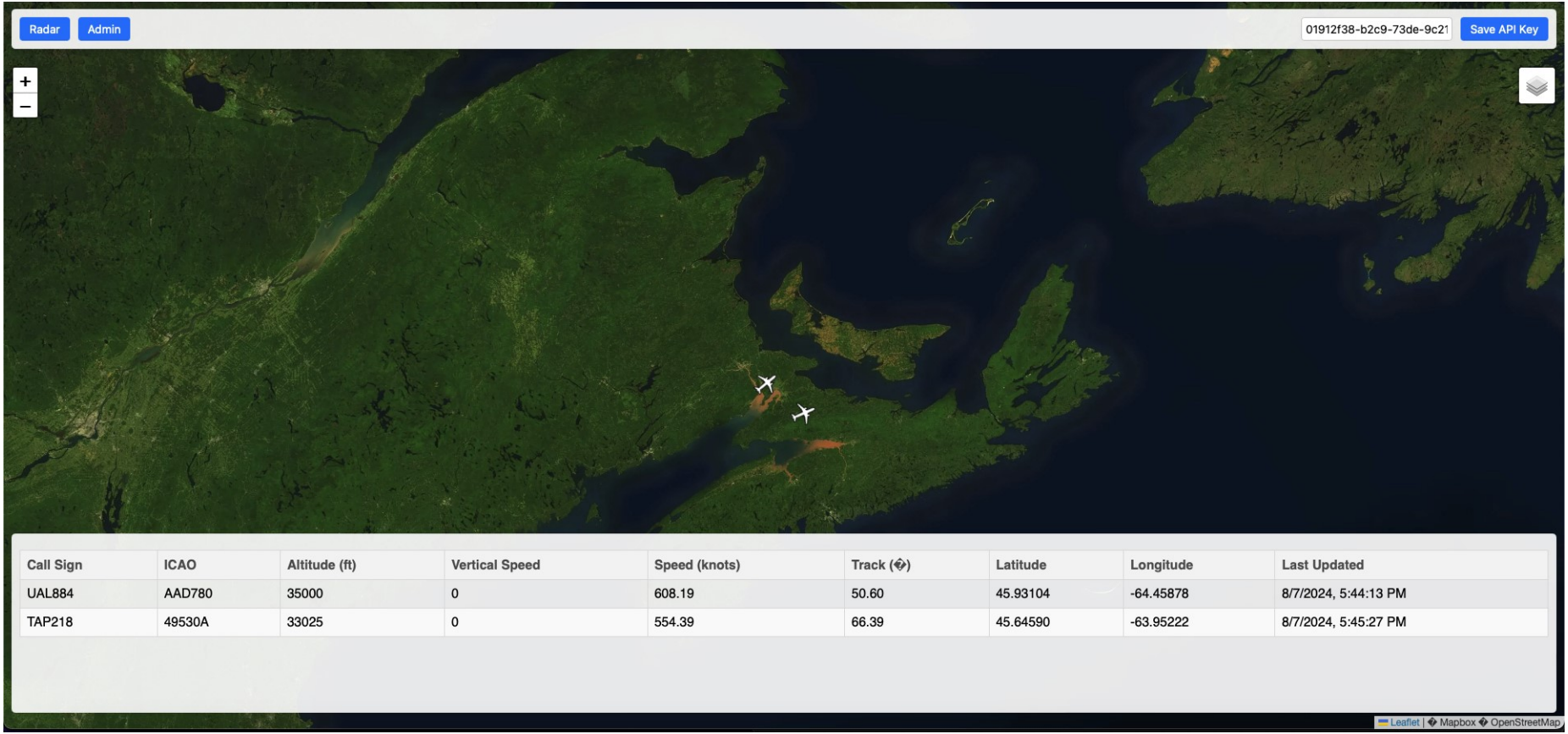
- Problem: A client in needed to run thousands of scenarios, each taking a non-trivial amount of time.
- Solution: Have one Jarvis act as a load-balancer for 20 other Jarvis instances, using HttpCommand to forward requests. Implemented a polling mechanism to see when results were ready.



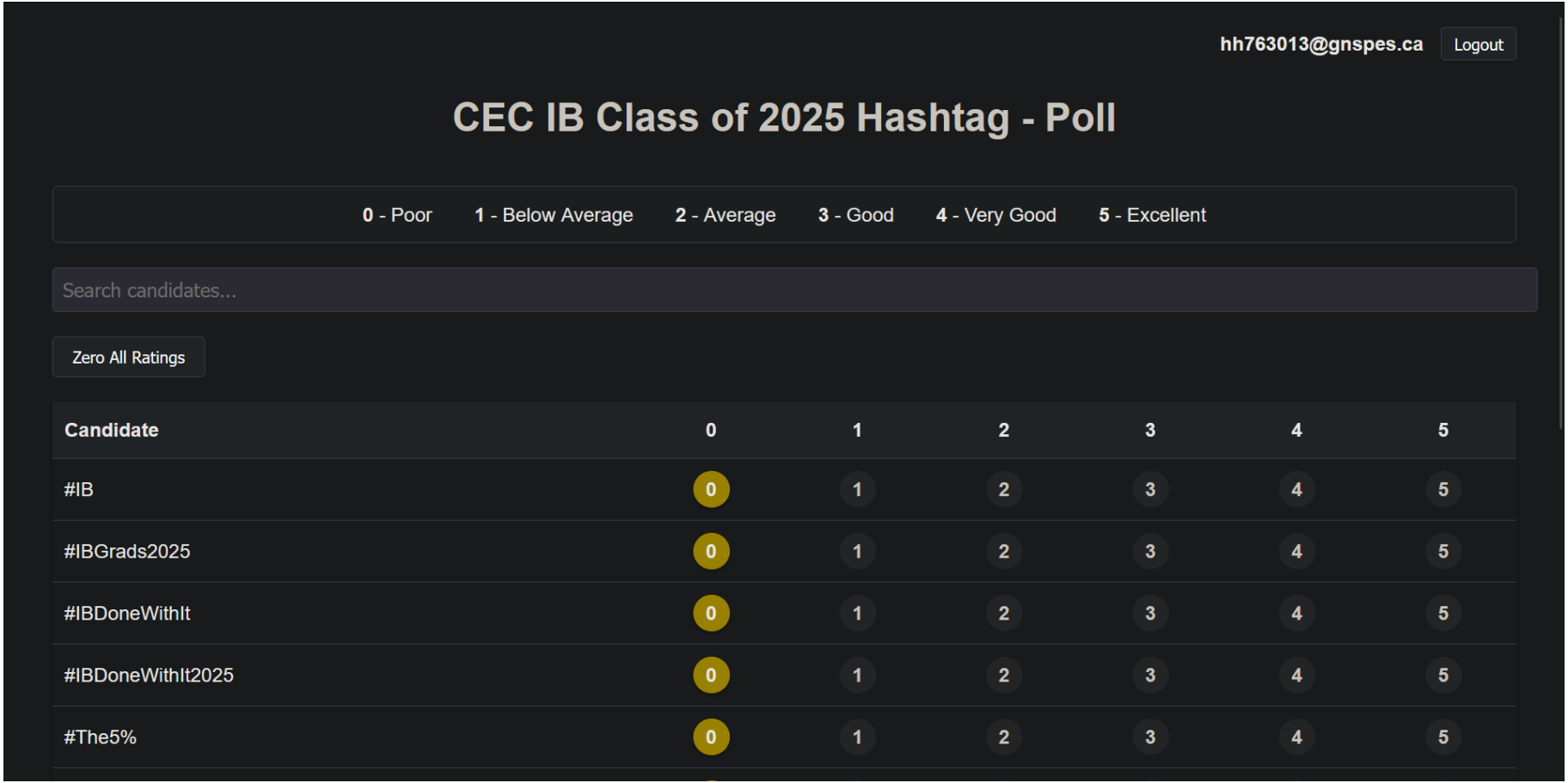
Holden Hoover

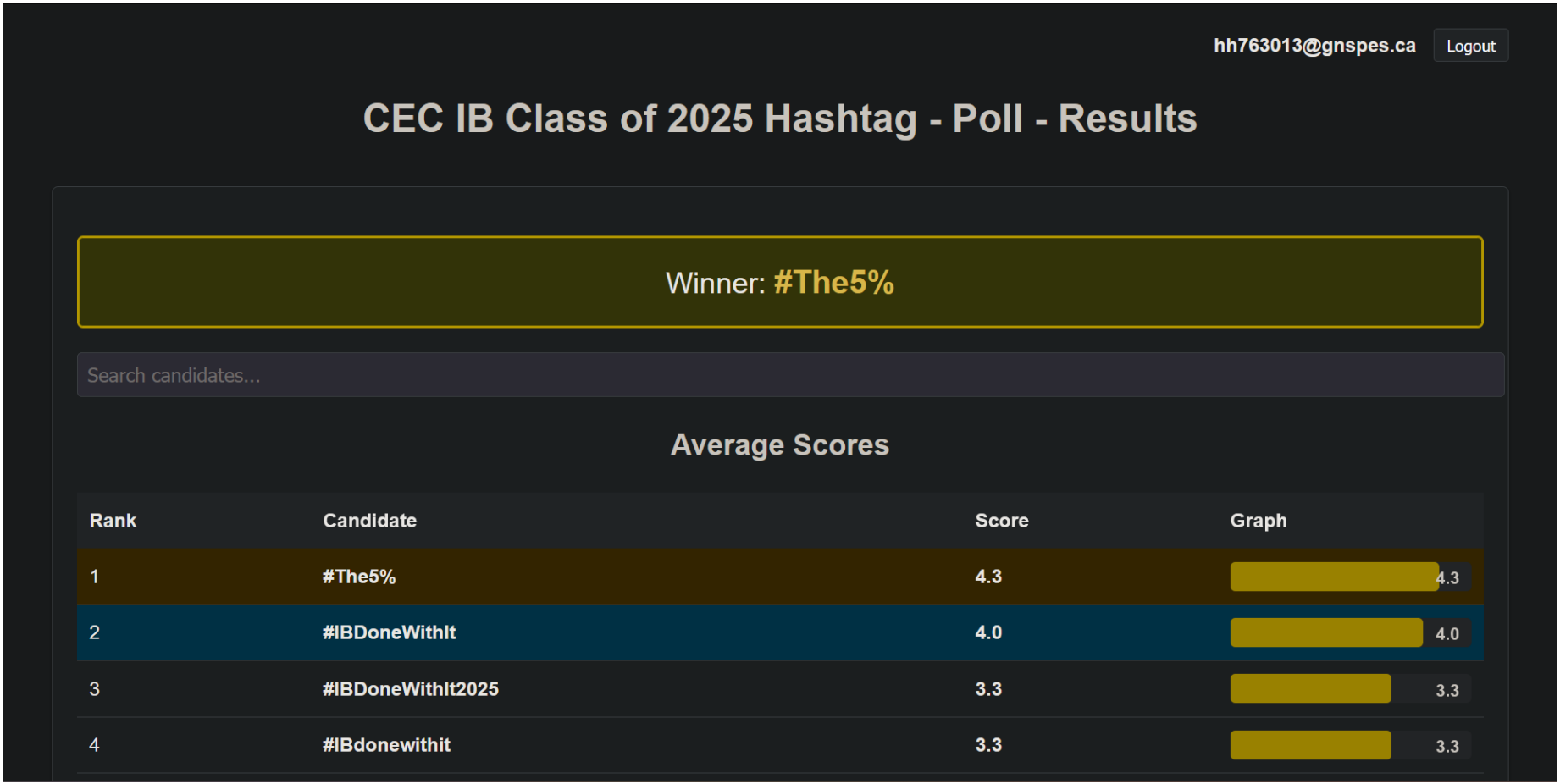
Holden has developed 2 Jarvis applications:

- His APL Forge winning Radar Ingest System (RIS) application that ingests airplane tracking data (emitted by airplanes in the air) from an array of antennas (that could be located around the world). This application used Jarvis for its GUI and Web API to view the data collected.
- STARAPL is an application he developed for a school voting system, to let people login with their school email accounts (Google Account OAuth) and then vote on various options (using the STAR Voting Method). After everyone voted, the results could then be calculated and displayed. STARAPL used Jarvis for the Web UI (which was available on the "public internet" while the poll was open). STARAPL also did have an API as well, for the Web UI to interact with.



Call Sign	ICAO	Altitude (ft)	Vertical Speed	Speed (knots)	Track (°)	Latitude	Longitude	Last Updated
UAL884	AAD780	35000	0	608.19	50.60	45.93104	-64.45878	8/7/2024, 5:44:13 PM
TAP218	49530A	33025	0	554.39	66.39	45.64590	-63.95222	8/7/2024, 5:45:27 PM

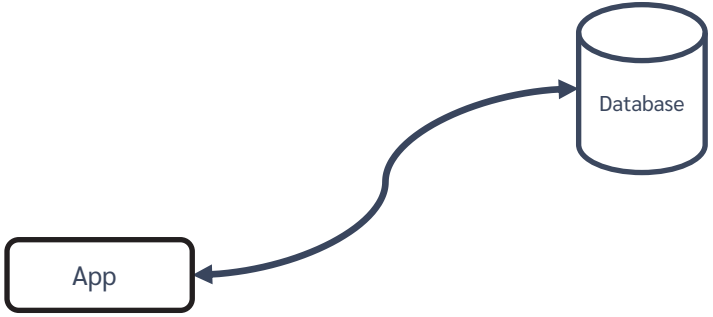

hh763013@gnspes.ca
Logout
CEC IB Class of 2025 Hashtag - Poll
0 - Poor
1 - Below Average
2 - Average
3 - Good
4 - Very Good
5 - Excellent
Search candidates...
Zero All Ratings
Candidate
0
1
2
3
4
5
#IB
#IBGrads2025
#IBDoneWithIt
#IBDoneWithIt2025
#The5%


hh763013@gnspes.ca
Logout
CEC IB Class of 2025 Hashtag - Poll - Results
Winner: #The5%
Search candidates...
Average Scores
Rank
Candidate
Score
Graph
1
#The5%
4.3
4.3
2
#IBDoneWithIt
4.0
4.0
3
#IBDoneWithIt2025
3.3
3.3
4
#IBdonewithit
3.3
3.3

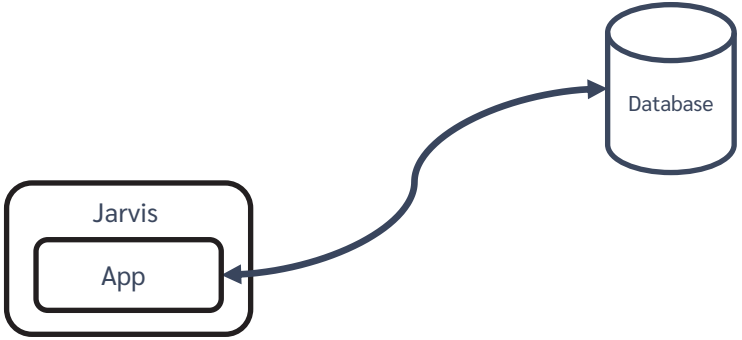
The Plan Visualized... (from Dyalog'22)

The Plan Visualized...

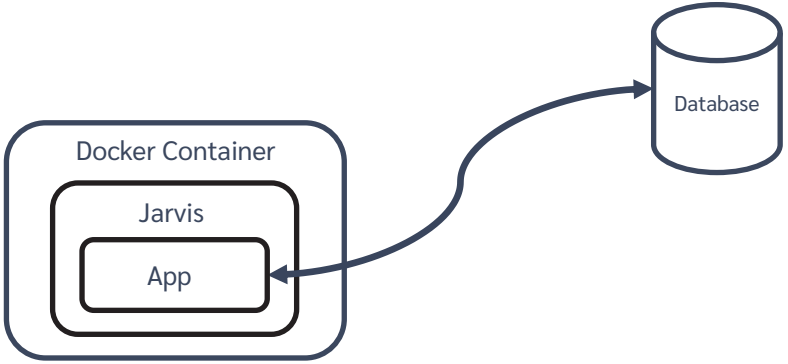
In the beginning, there was an Application...



Run the app as a service

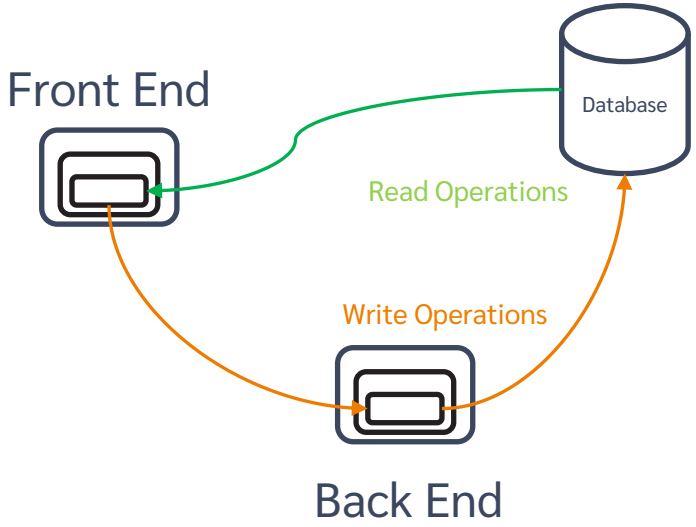


Run it in a container

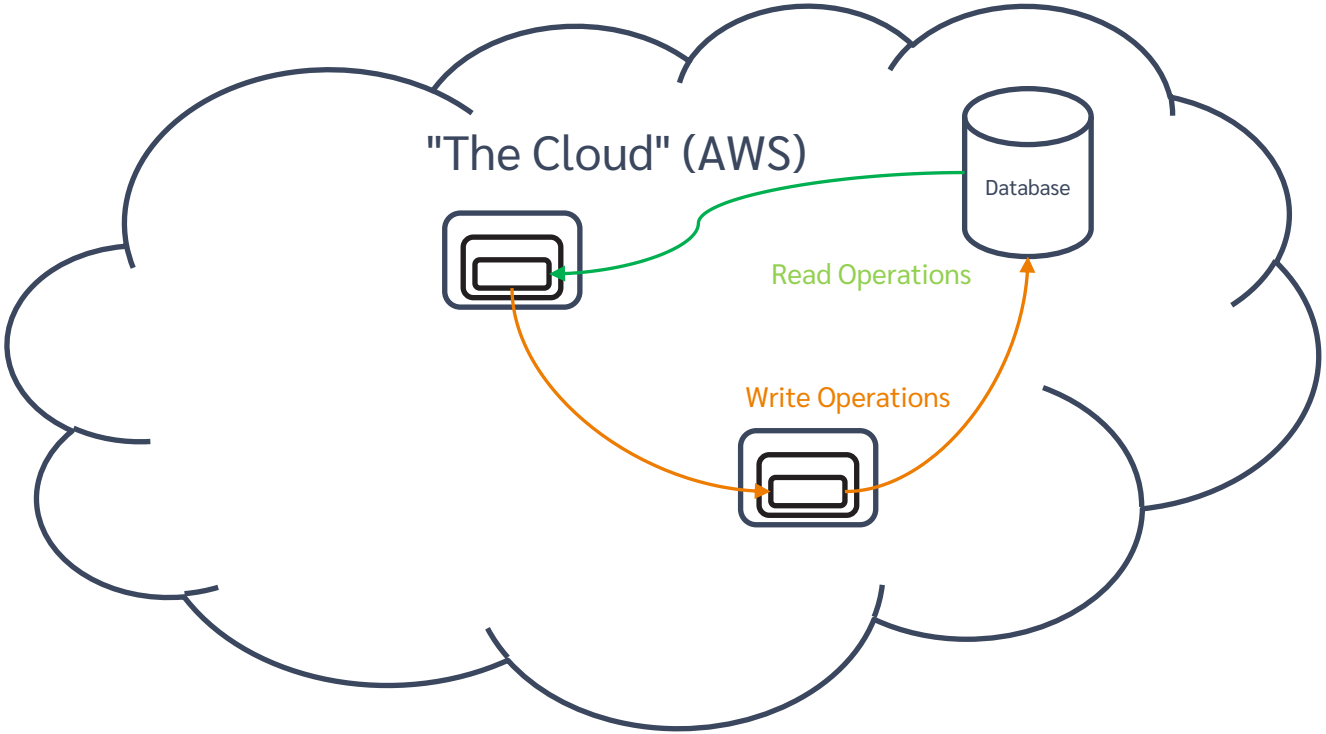


Split into Front and Back Ends

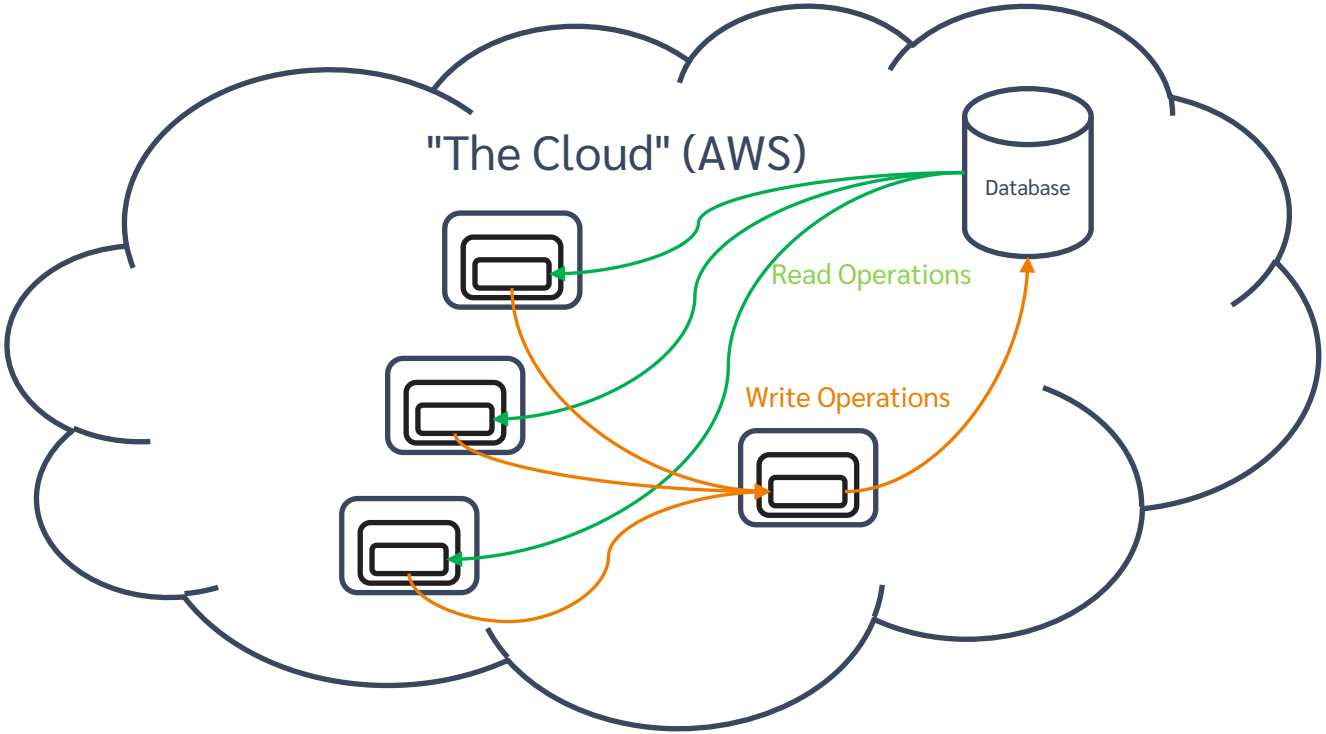
We'll call this "Two-Tier"



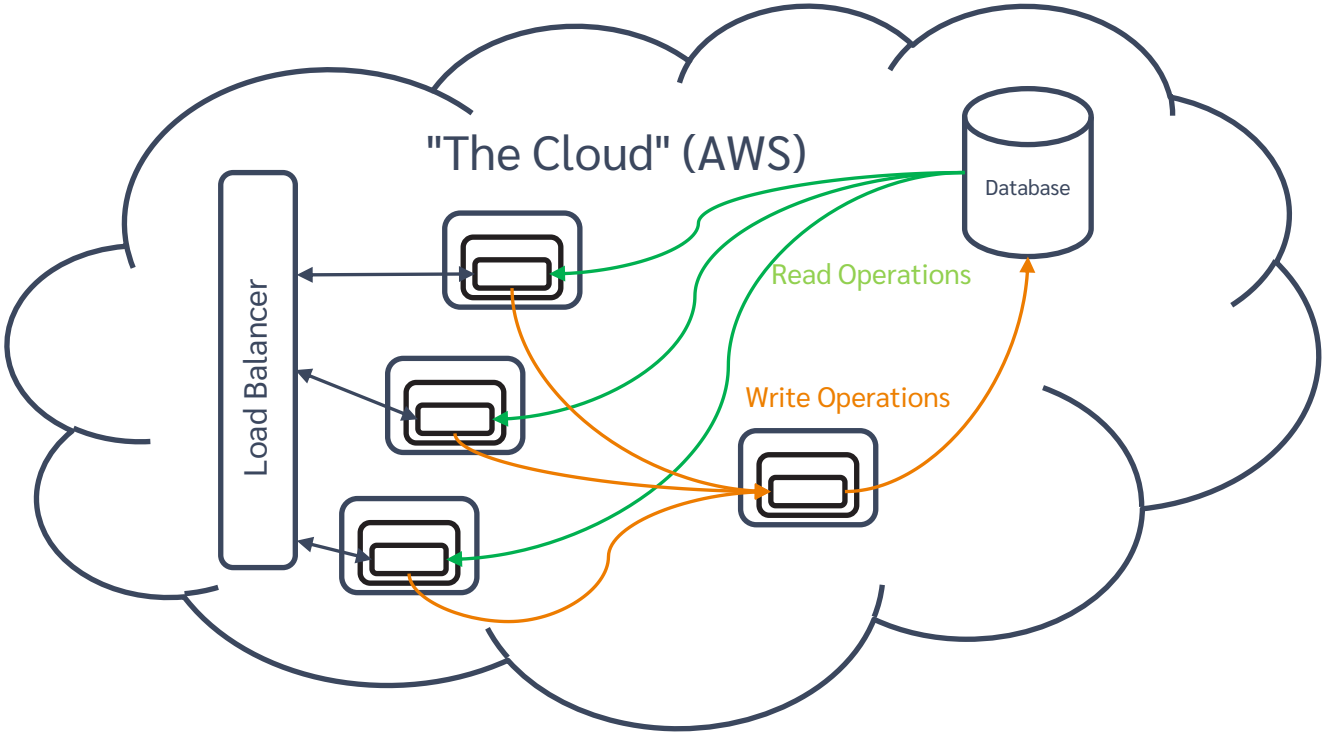
Try it in the cloud



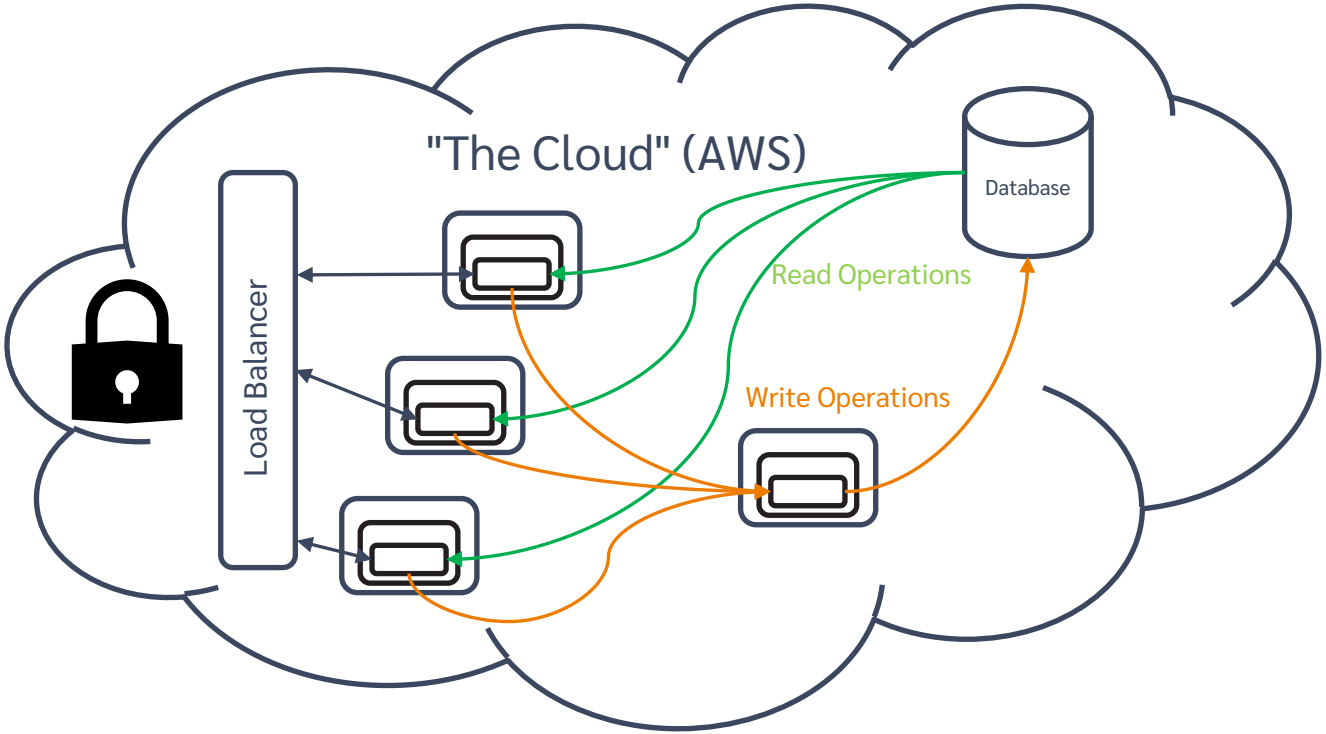
Scale it up



Load balance it



Secure it


"The Cloud" (AWS)
Load Balancer
Database
Read Operations
Write Operations

Takeaways

- In general people have found Jarvis flexible and easy to use
- The core design has held up – still at major version 1.
- If you need it to do something – ask!