ENGAGE
user group

Breaking the Monolithic: Architecting Microapps on HCL Domino

Richard Moy - April 21th, 2026

About Me

- Managing Director of Phora Group
- Life-time IBM Champion
- Life-time HCL Ambassador
- Working with Notes since 1995
- Still a big fan of Dojo Toolkit
- Coordinator for CollabSphere for 15 years
- Grandpa
- Hiking and biking enthusiast

About Phora Group

- HCL/IBM Business Partner
- HCL Domino consultants since 2003
- Business process automation services since 2009
- Primary focused on Web-based solutions
- Over 60+ years of combined experience in Notes/Domino

About Phora Group

Creators of the iPhora family of Domino-based products

- iPhora AppBuilder – No code application builder
- iPhora AppPlace – Place-based platform to distribute and run apps
- iPhora Automate – No code process automation platform

iPhora Automate

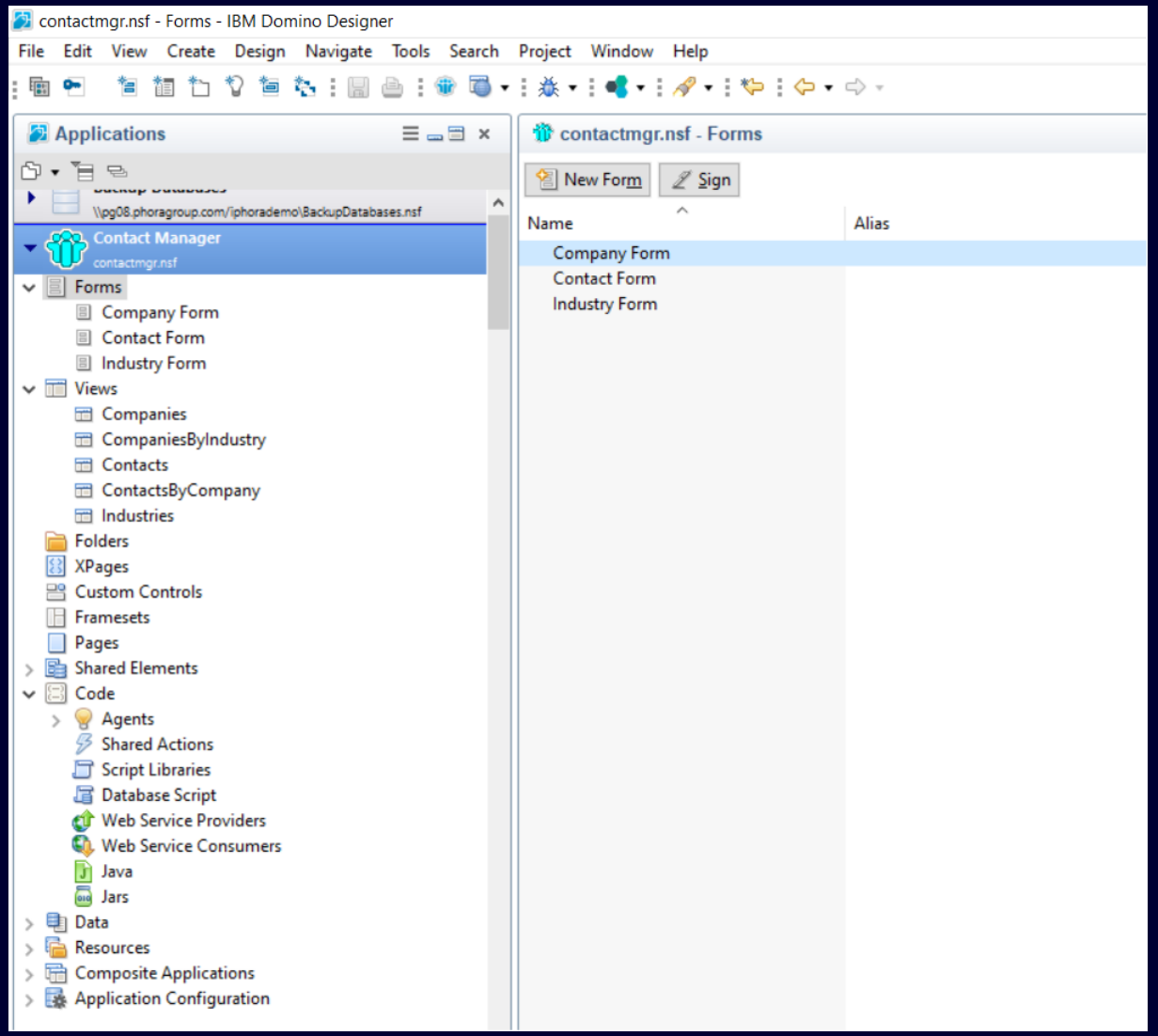
- Try iPhora Automate on https://iphora.io
- New released version, v2026043001.

Agenda

- Building Domino Apps
- Monolithic Apps
- What are Microapps?
- Our Use Case
- HCL Domino Challenges
- Creating a Micro App on HCL Domino
- Demo
- Summary

Creating a Typical Domino Applications

Contacts



Building a Domino App

- Forms
- Views
- Pages
- Agents
- Navigators
- Java
- LotusScript
- Formula Language
- JavaScript

Building a Domino App

- Most start off simple
- Add more and more features into a single Domino database
- Requires more and more modifications to hold things together
- Splitting up the applications to multiple databases
- Add code to connect to external services
- More and more patches are added to make things work.
- Making it become monolithic

Challenges of Monolithic Apps

- Complexity
- Changes become more and more difficult and slower
- Integration of new technologies is harder
- Scalability
- Deployment and reliability
- Usability

Our Microapp Use Case

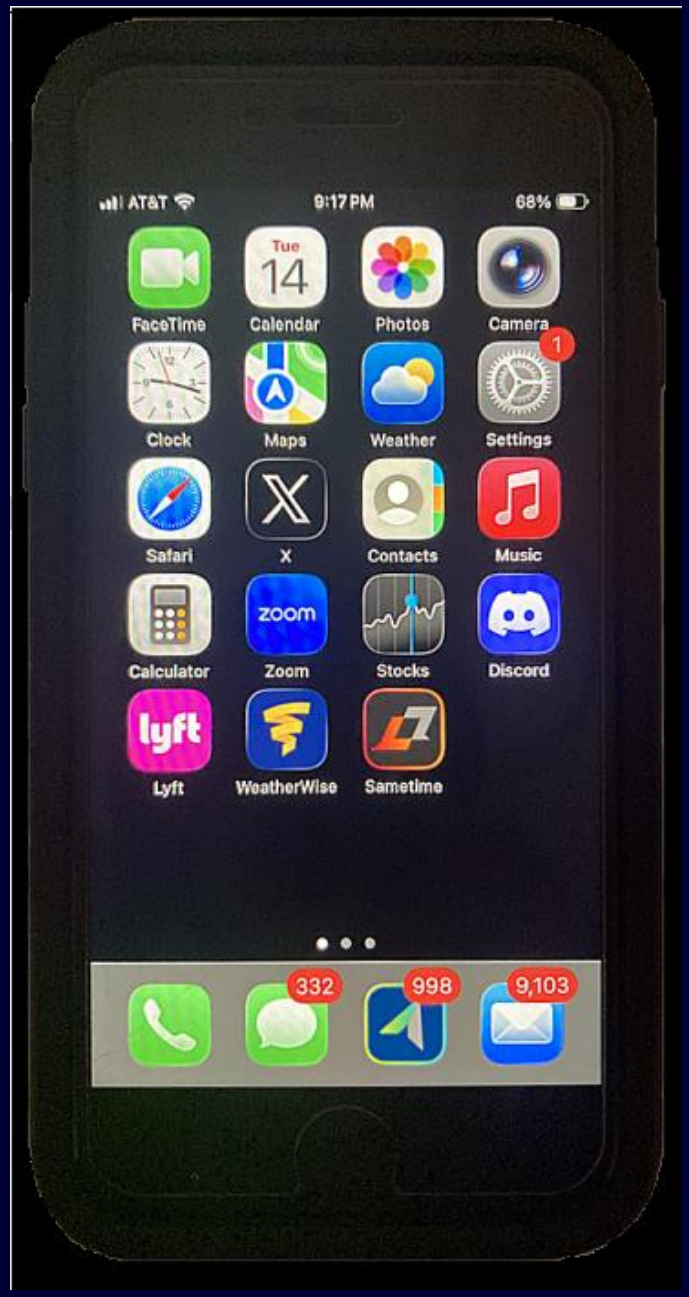
In designing and building our series of products, we found:

- App creation by individual business users
- Ability for these apps to be plug and play
- Ability for apps to share data and services
- Deployment of apps via an app store
- Scalable solutions
- Simple consistent user interfaces throughout apps

The Rise of Mobile Apps

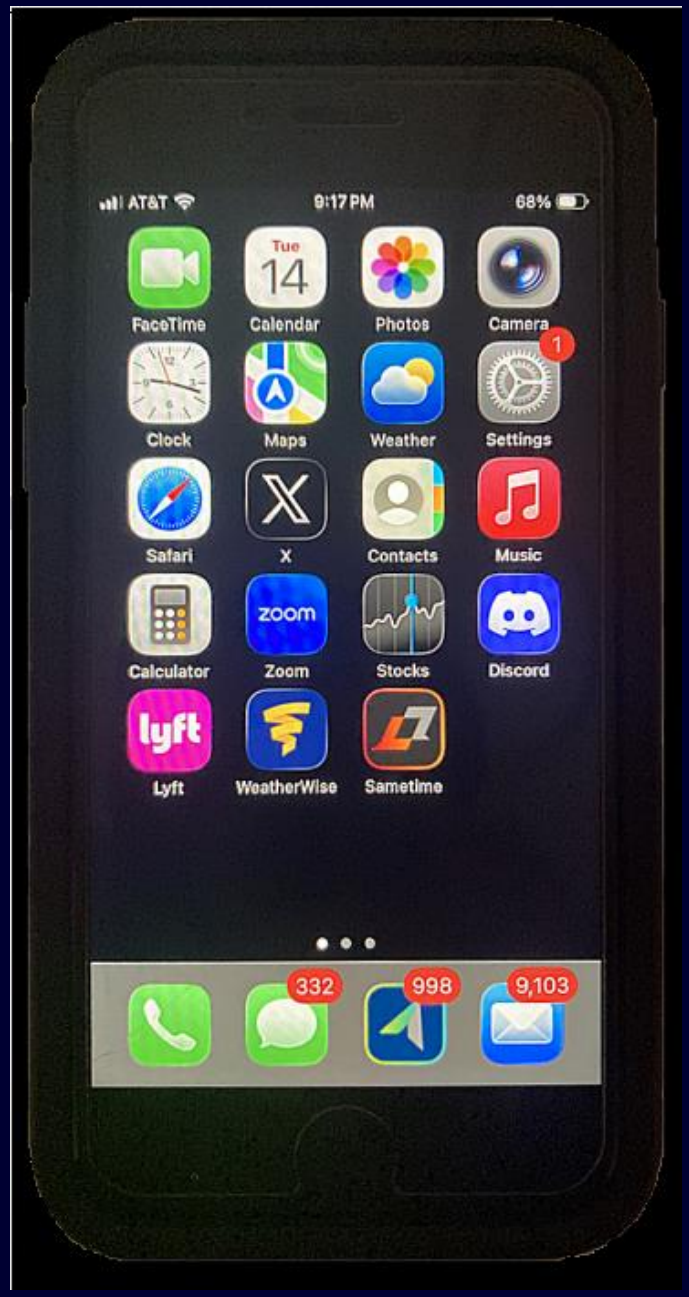
- 1997 - the first mobile app from Nokia
- 2007 – Apple App Store
- 2026 - 4 Million+ Mobile apps

Mobile apps are generally very task specific and maintains a consistent and intuitive user interface.



Genesis of Microapps

- Businesses wanted to apply similar design to their enterprise applications.
- Businesses wanted the simplified approach of separating the functionality into different apps
- Businesses wanted to incorporate the same intuitive user interfaces.
- Businesses identified the effective development approach that lessen the need for support.



What are Microapps?

- Autonomous applications are built for a single purpose.
- Each microapp performs a set of focused task
- Provide a consistent user experience throughout the apps
- Disposable
- Web-based
- Notification driven

Microapp Growth Trend

Popularity of microapps is driven by vibe coding and no-coding where users can create their own app

Key Benefits of Micro Apps

- Faster development and deployment
- Improved user focus and adoption
- Easier maintenance and updates
- Integration with enterprise systems
- Scalability and flexibility
- Can be created by the “consumer”

Consumer vs Enterprise Microapps

Consumer

- “Disposable” including data
- Microapps are mostly standalone
- Need to inhere to the need of the individual consumer
- Localize security
- Individual productivity
- Individual creator/contributor

Enterprise

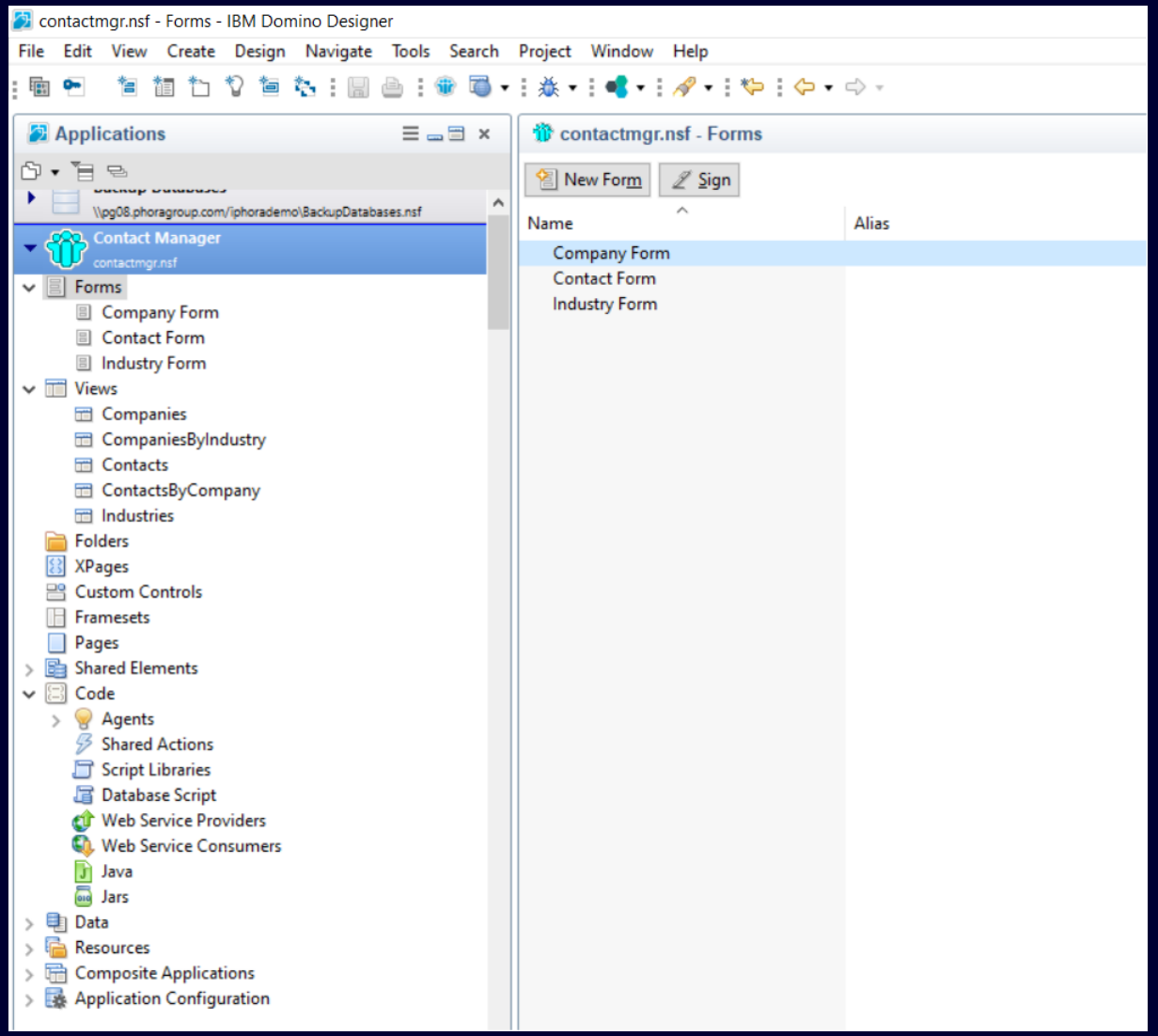
- **“Disposable” but not data**
- Microapps need to integrated with other microapps and enterprise solutions
- Need to inhere to corporate governance
- Enterprise security
- Multiple users
- Multiple creators/contributors

Attributes of Enterprise Microapps

- Pluggable API-led development
- Security and control
- Deployment via an app store
- Microservices-based systems
- Event bus driven
- Automated low-code/no-code development

Lotus Notes – First Microapp platform

- User could create apps
- Targeted single purpose task
- Security and control
- Application template



Challenges in Building Microapps on Domino

- No out of the box ability for creating apps using no-code / low-code / vibe coding
- No consistent framework for web-based apps
- Domino apps in general are not API driven platform
- Domino does not support a microservice architecture
- Domino does not provide an event bus architecture

Advantages in Building Apps on Domino

- Full built-in enterprise security
- Full control of data (private and sovereign)
- Flexible application environment
- Domino databases are inherently container-based

Creating a Domino Microapp

To create enterprise microapps on Domino we need to have:

- User driven application development
- Plug and play API driven architecture
- Microservices architecture
- Event Bus

Creating Domino Microapp - UI

Domino provides a flexible app container where you have many ways of creating the UI

- No-code tools
- Vibe coding
- Domino Leap (limited)

Creating Domino Microapp – API Driven

HCL Domino allows you to build APIs

- LotusScript/Java Agents
- Java
- Xpage agents
- DAS

Creating Domino Microapp - Microservices

Domino does not support microservices. However, it does have:

- Run on server agents
- Scheduled agents

Creating Domino Microapps – Event Bus

openntf Webinar

Creating a Service Oriented Architecture (SOA) Platform with Domino

October 16, 2025

openntf

Enterprise Event Bus

SOA SERVICE (Microservice)

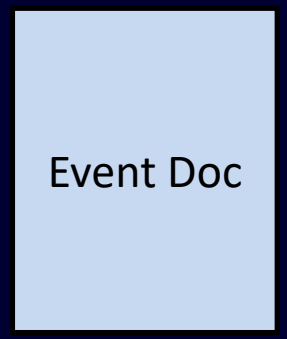
RUN ON SERVER AGENT (LOOSELY)

Services:
Reusable components that each performing a distinct business function.

Loose coupling:
Services are designed to be independent

Standardization:
Standard interfaces and protocols


ESB NSF


Event Doc


Payload
Doc


Transport
Doc

Event Document: Service Chain

```
[{
        "id":"apple",
        "server": "ACME01/ACME1",
        "db": "mydb",
        “esb”:”mailin1”,
        "service": "myagent1",
        "async": "no",
        "data": {
                "id": "a",
                "label": "b"
        },
        "next_id":"orange"
}, {
        "id":"orange",
        "server": "ACME02/ACME2",
        "db": "my2db",
        “esb”:”mailin1”,
        "service": "myagent2",
        "async": "yes",
        "data": {
                "id": "a",
                "label": "b"
        }
        "next_id":""
}]
```

Transport Document: Memo

Fields

subject:	universal ID of the payload document
server:	name of the targeted server/domain
database:	directory path of the targeted database that has the ROSA
agent:	name of the targeted ROSA

Payload Document: Posted Data

```
{
        “firstname":“John",
        “lastname”:”Smith”,
        “company”:”Acme Corporation”,
        “email”: johnsmith@acme.com,
        “service”:”myagent”,
        “serviceid”:”apple”,
        “eventid”:”event1”
}
```

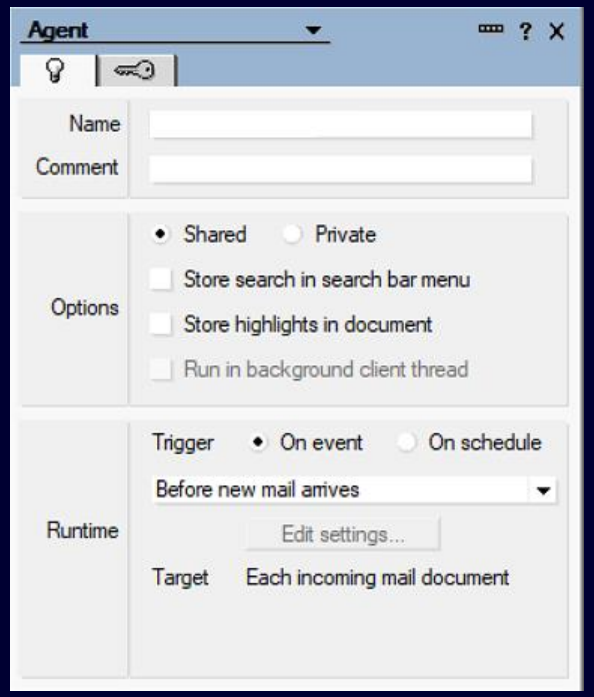
Analogous to a Domino agent parameterDoc

Triggered By Before Mail Arrives Events

PUBLISHING ROSA

NON-BLOCKING,
ASYNCHRONOUS PROCESS

- On Event agent
- Creates payload document
- Run targeted schedule agent using command line shell



Call session.SendConsoleCommand(TS ,|tell amgr run "| & TP & |" '| & ROSA & |'|)

Information found in the transport document

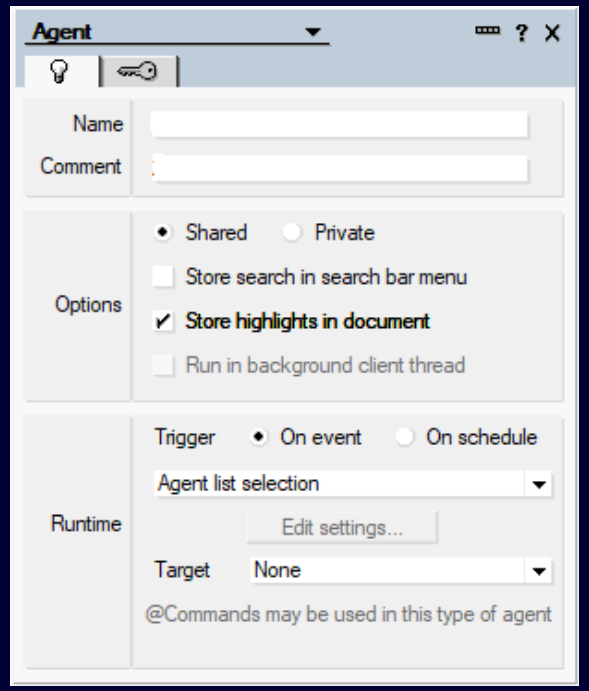
TS: target server
TP: target database path
ROSA: Name of the run on server agent

Non-Blocking, Asynchronous Process

SUBCRIBING ROSA

"async":"yes"

- Scheduled Agent
- Scans in Event Bus View to find all payload documents that it needs to process that is assigned to itself
- Reads the payload document for the submitted data
- Runs Business Logic for service



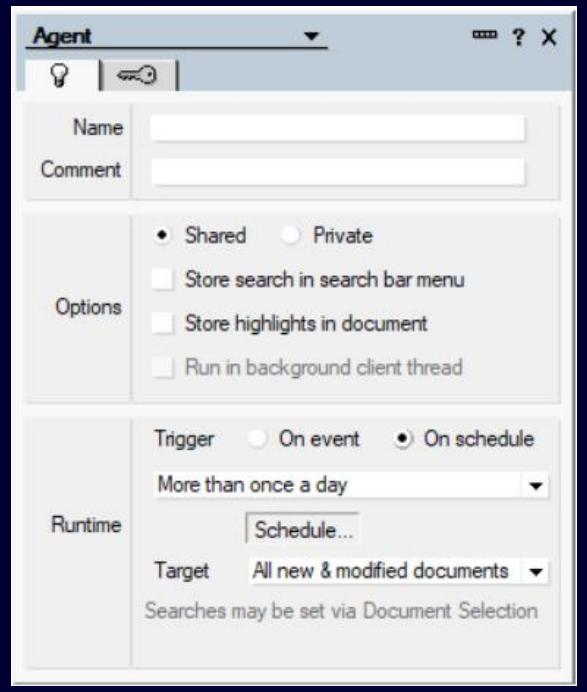
Blocking, Synchronous Process

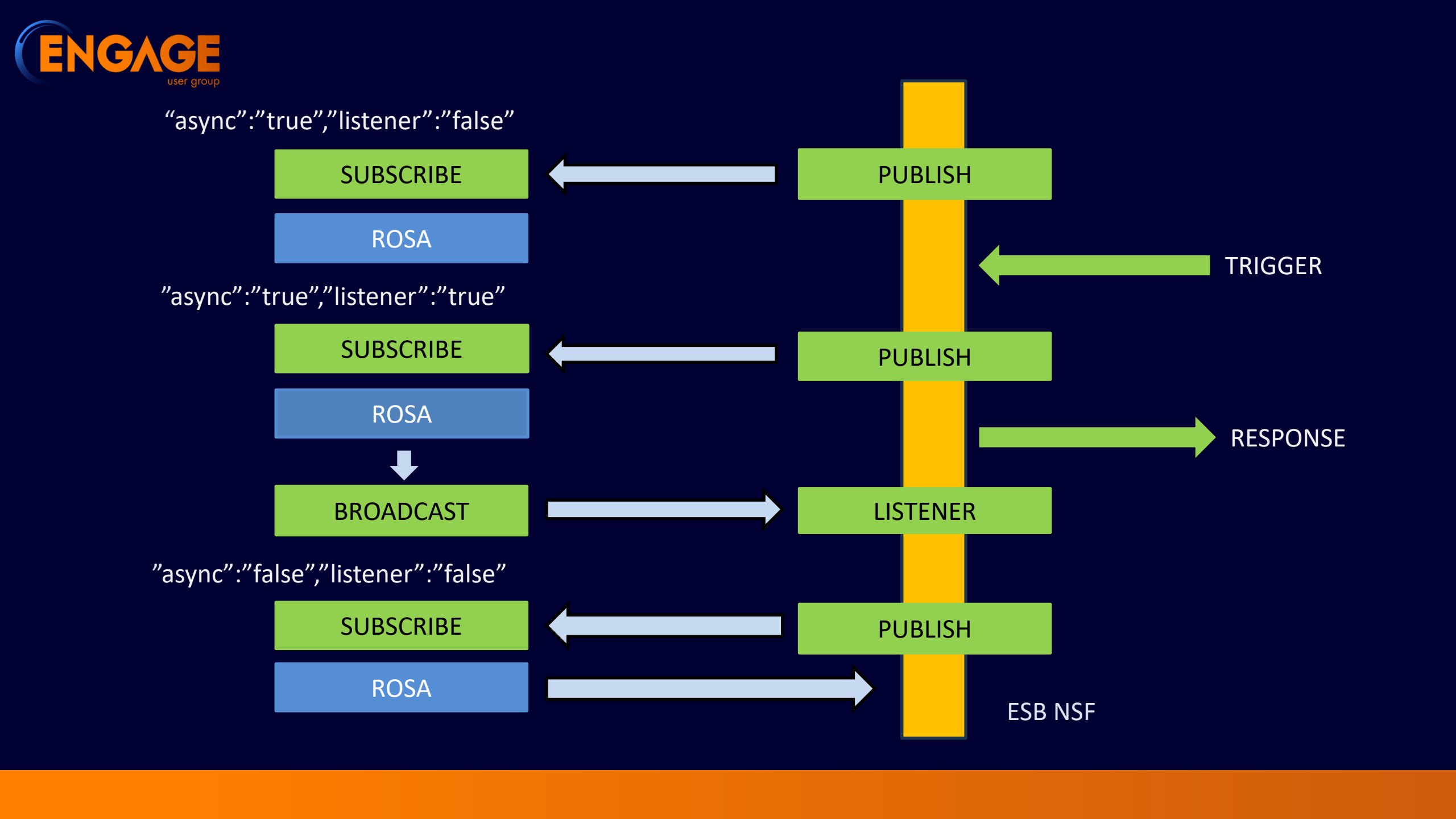
SUBCRIBING ROSA

“async”:”no”

- Scheduled Agent
- Scans in Event Bus View to find any request that it needs to process
- Reads the payload document for the submitted data
- Runs Business Logic for service

- Get next service in the service chain from event document
- Creates payload document
- Creates transport document (memo)
- Sends message to ESB (mail-in db)



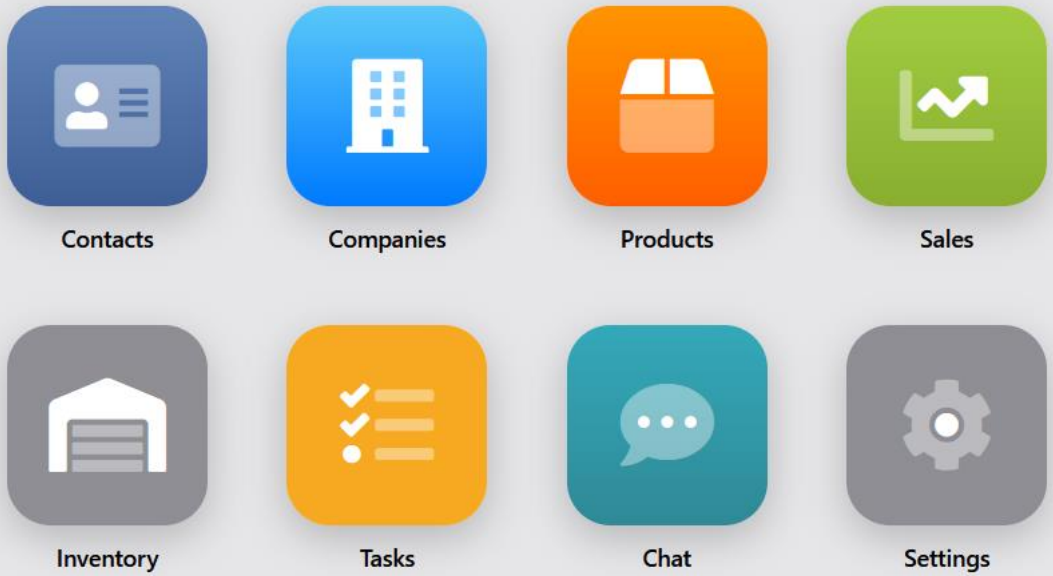

“async”:”true”,”listener”:”false”
SUBSCRIBE
ROSA
PUBLISH
TRIGGER
”async”:”true”,”listener”:”true”
SUBSCRIBE
ROSA
PUBLISH
RESPONSE
BROADCAST
LISTENER
”async”:”false”,”listener”:”false”
SUBSCRIBE
ROSA
PUBLISH
ESB NSF

ENGAGE
user group

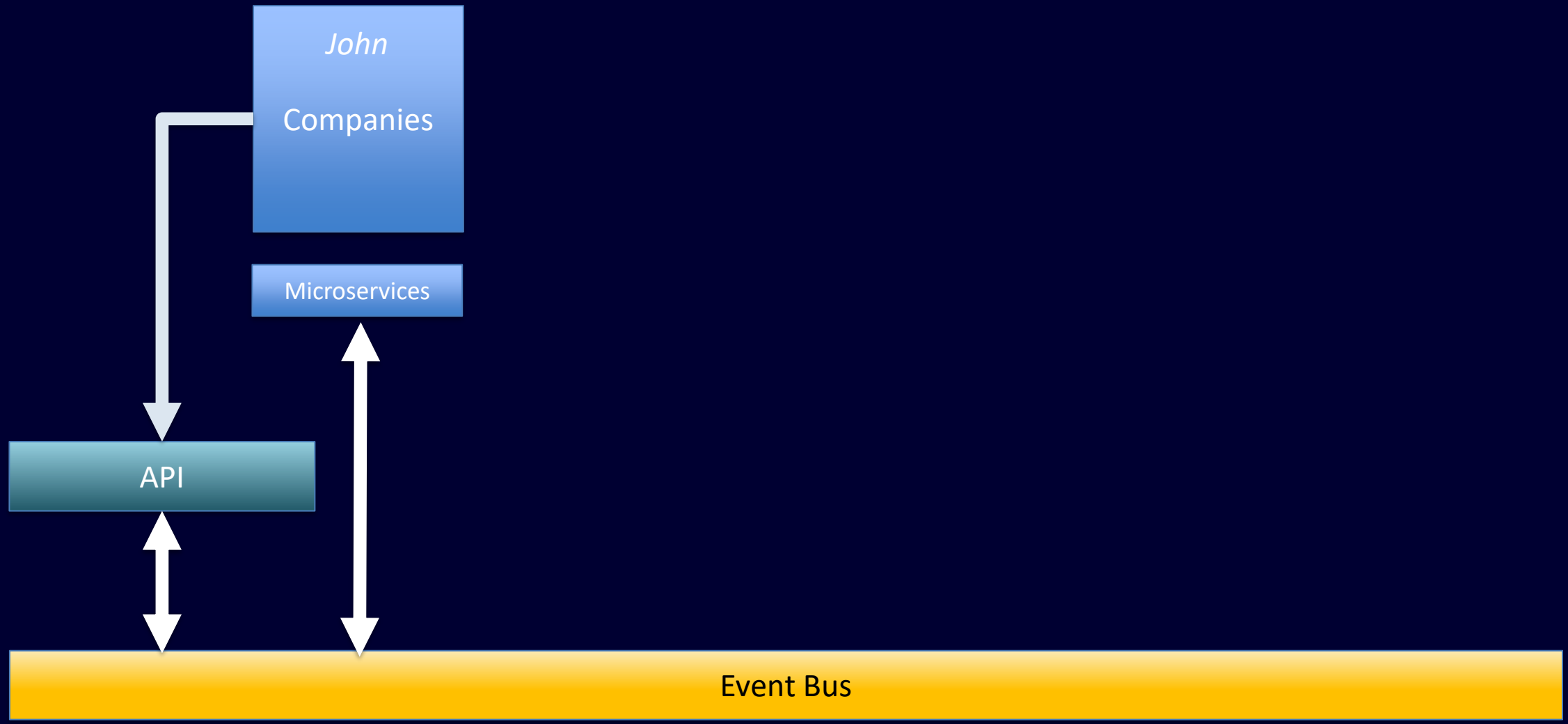
DEMO

Creating Demo Microapp UI

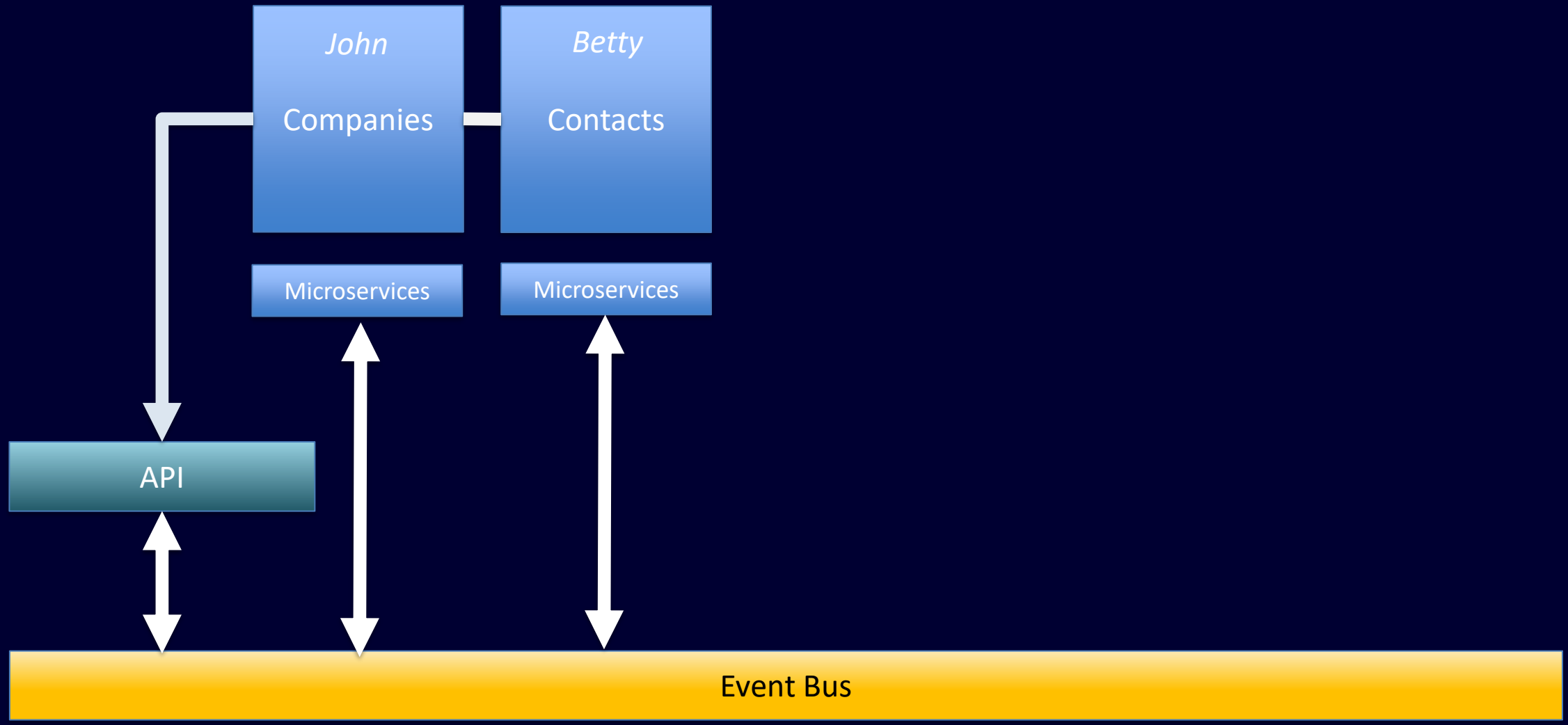
- Vibe coded 85% user interface – 2 hours on Gemini 3 Pro
- 15% modifications to make it work and fix mistakes – 2 days
- No framework - plain ES6 JavaScript and CSS


Contacts
Companies
Products
Sales
Inventory
Tasks
Chat
Settings

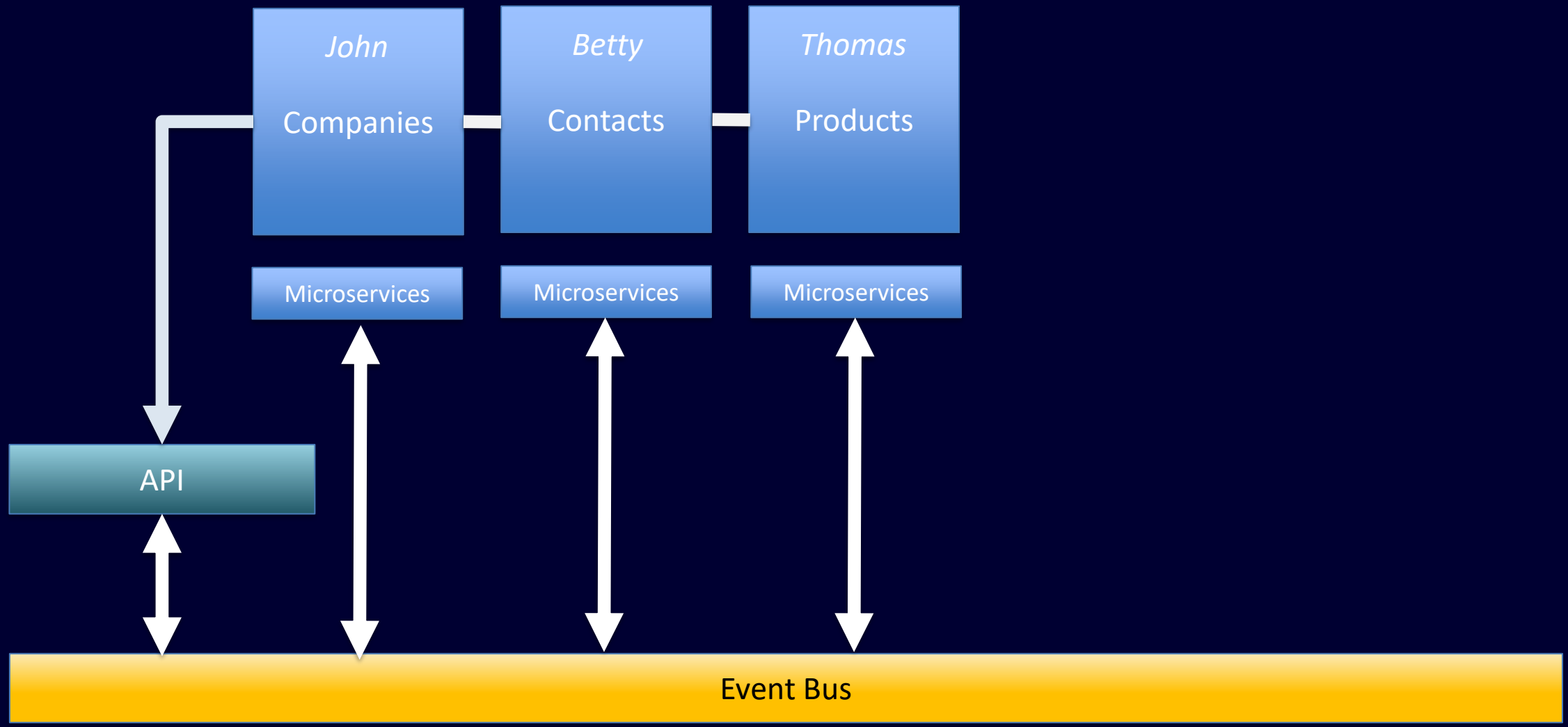
Companies App



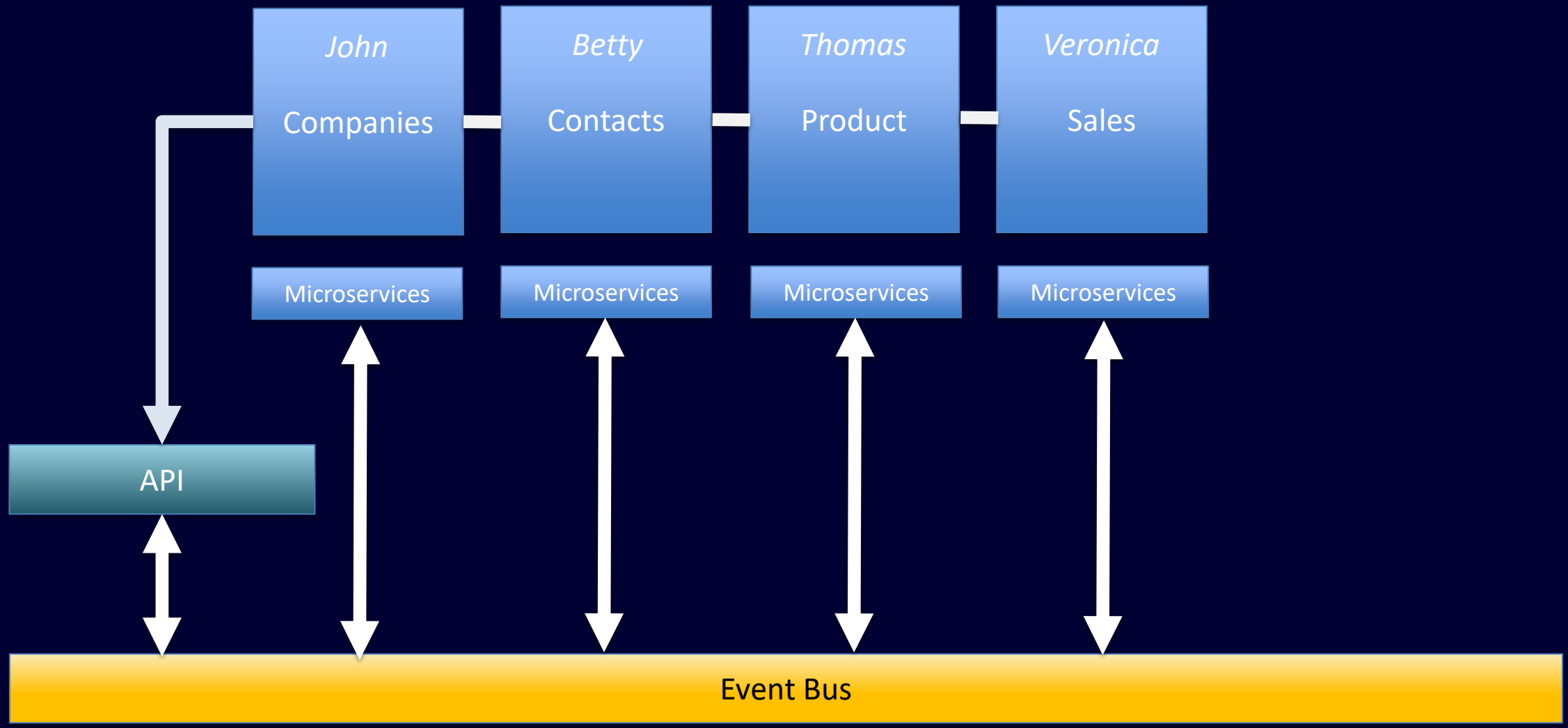
Contacts App



Add a Products App

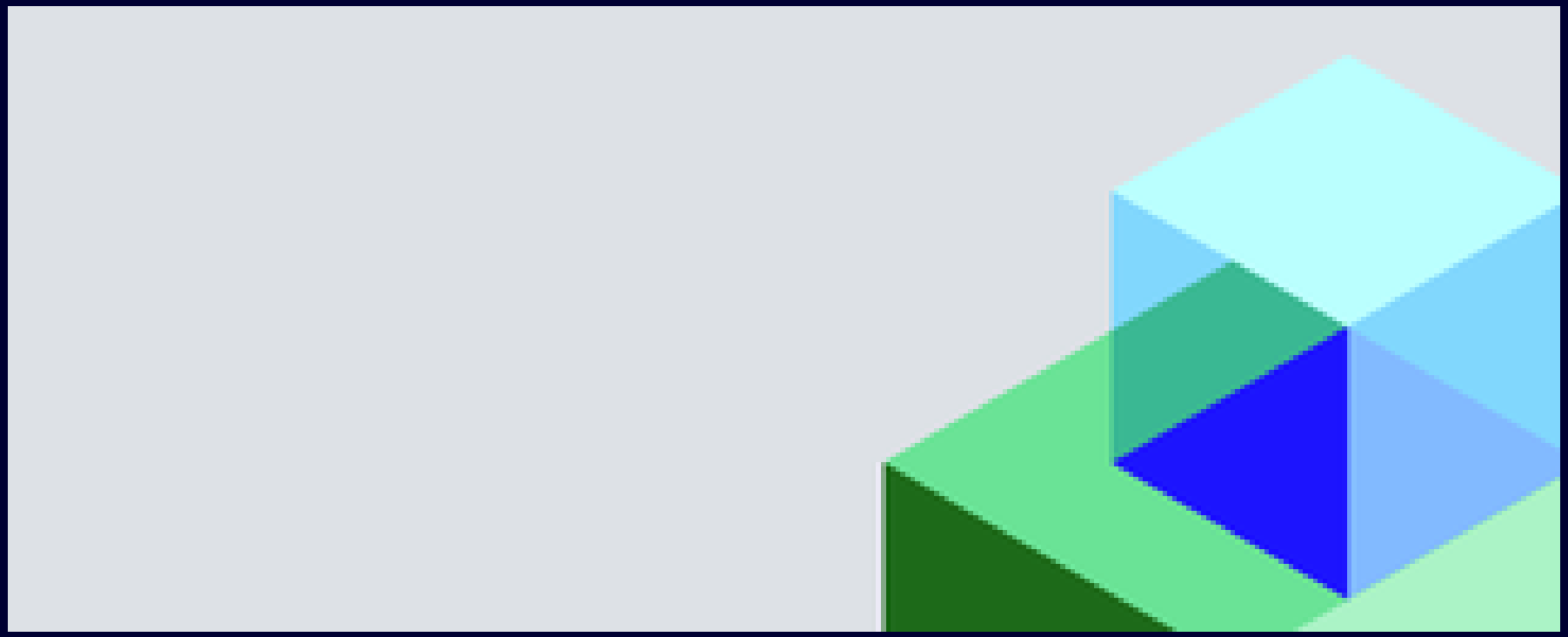


Add a Sales App

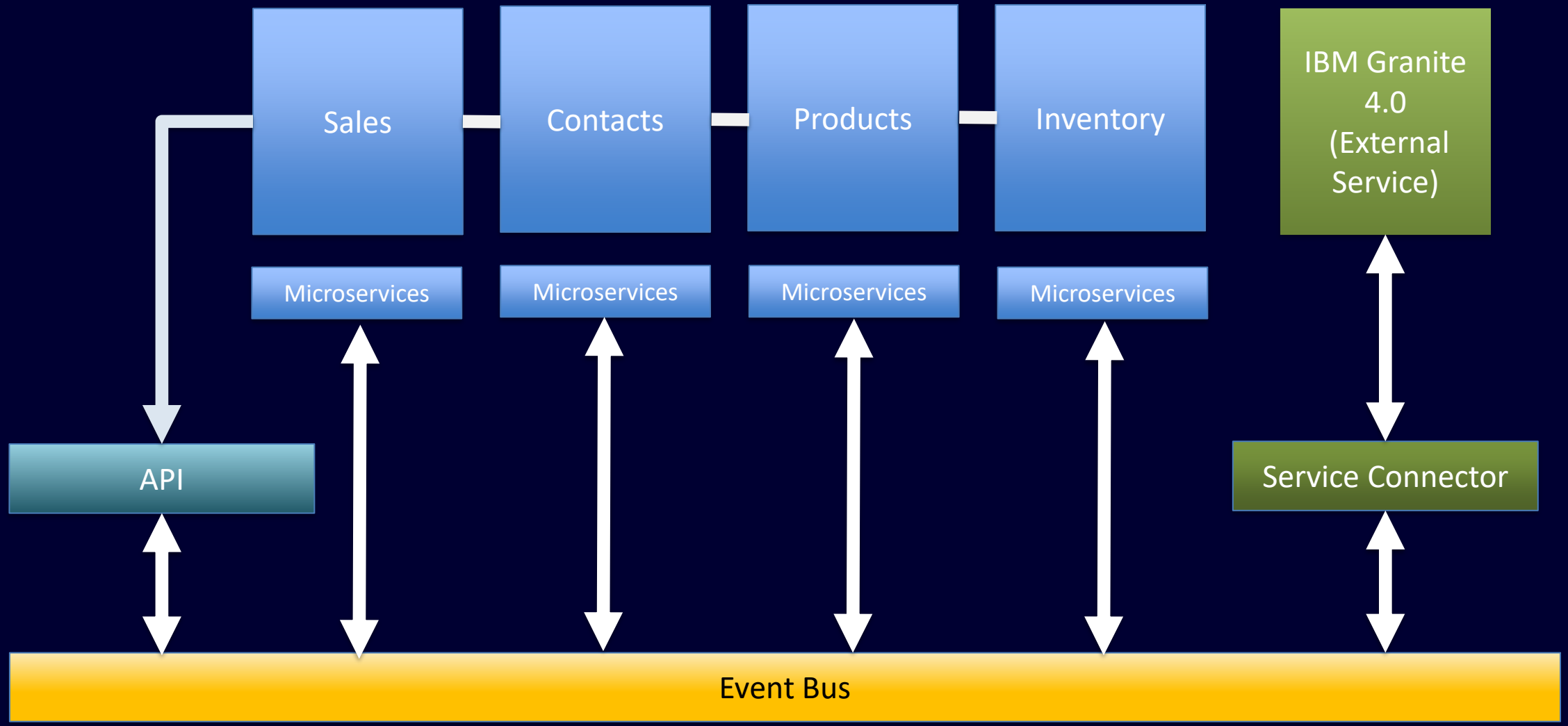


ENGAGE user group

IBM Granite 4.0: hyper-efficient, high performance hybrid models for enterprise



Sales App

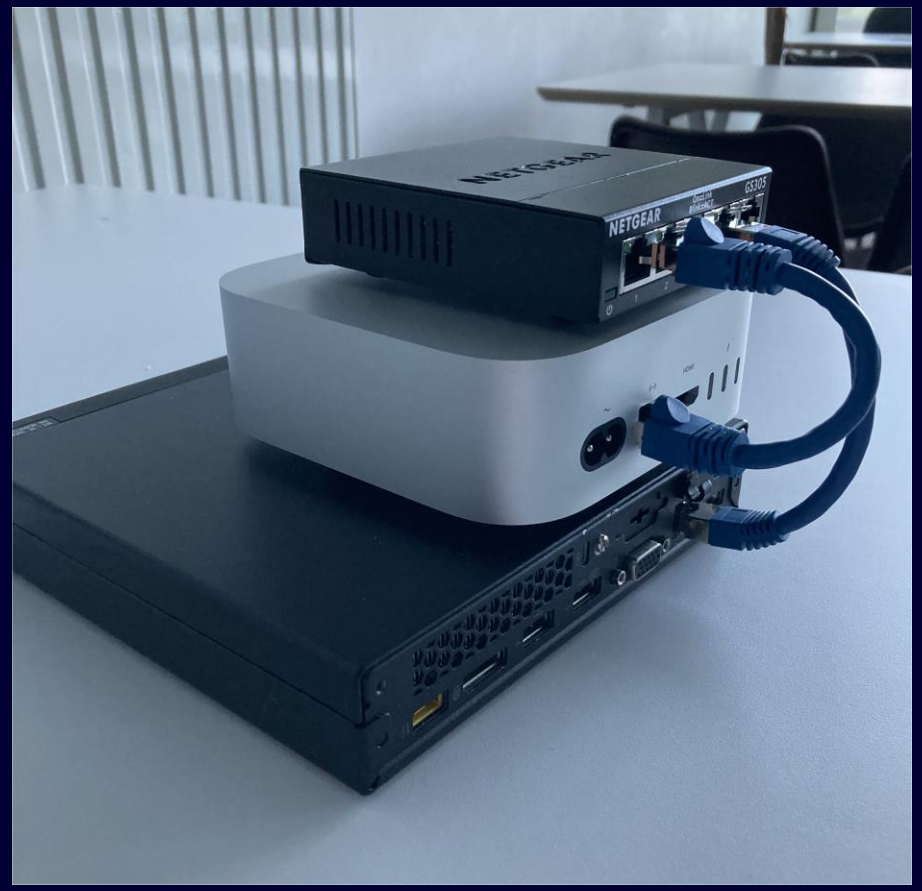

Sales
Contacts
Products
Inventory
IBM Granite 4.0 (External Service)
Microservices
Microservices
Microservices
Microservices
API
Service Connector
Event Bus

Domino-based Microapp Architecture

Domino Microapp Contains:

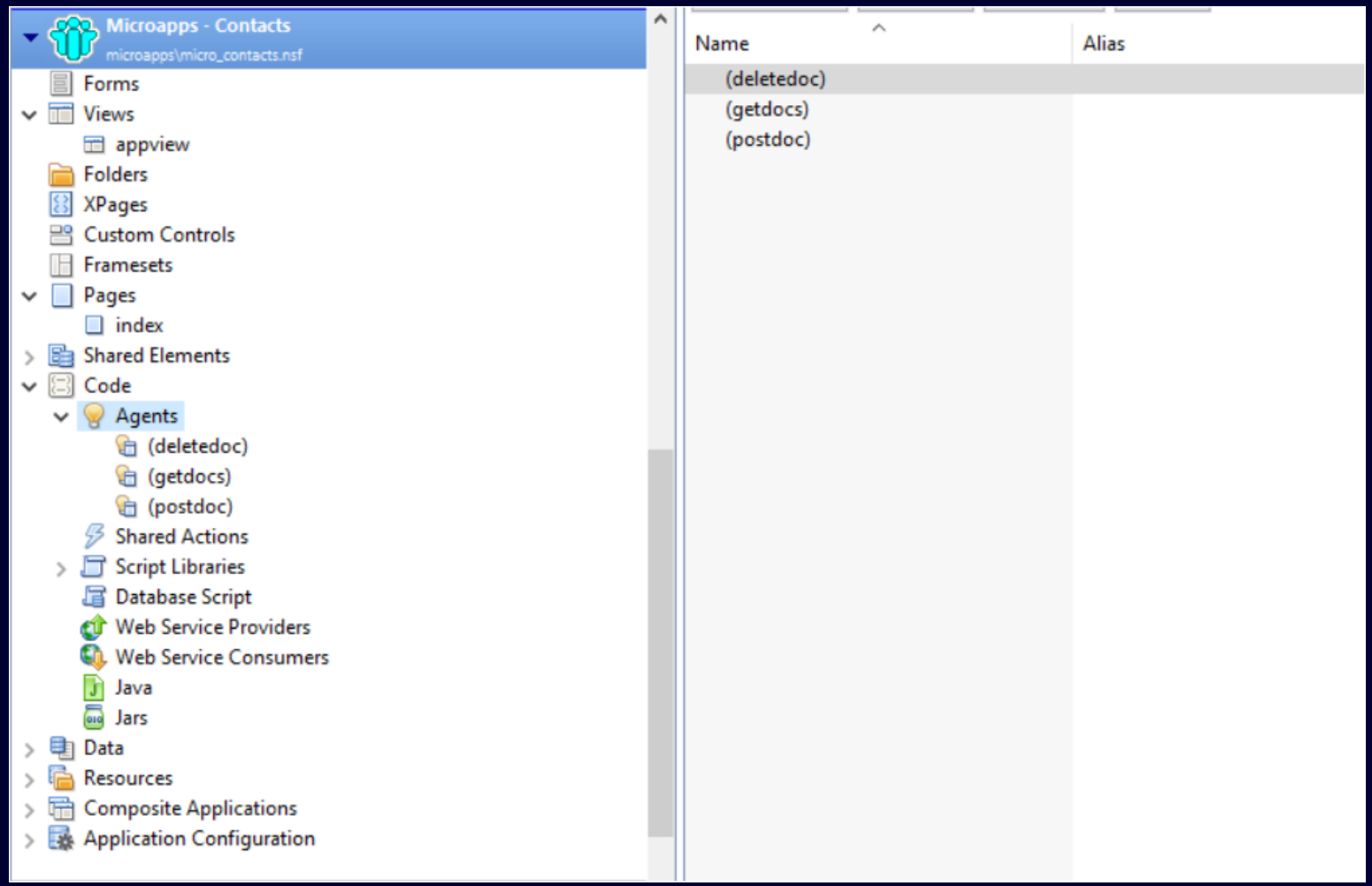
- Single Web Page
- One Domino Page to hold the UI
- No Domino Views (Not Needed by Required by Domino)
- Page specific JavaScript code
- App specific services , synchronous and asynchronous (Run On Server agents)

Demo Setup



- Lenovo, ThinkCentre
- Docker 29.3.1
- HCL Domino 12.02 FP7
- Mac Mini 4, Ollama 0.20.7 server
- IBM Granite4:7b-a1b-h model
- Microapp using Classic Domino Web Development

Domino-based Microapp Architecture



Domino-based Microapp Architecture

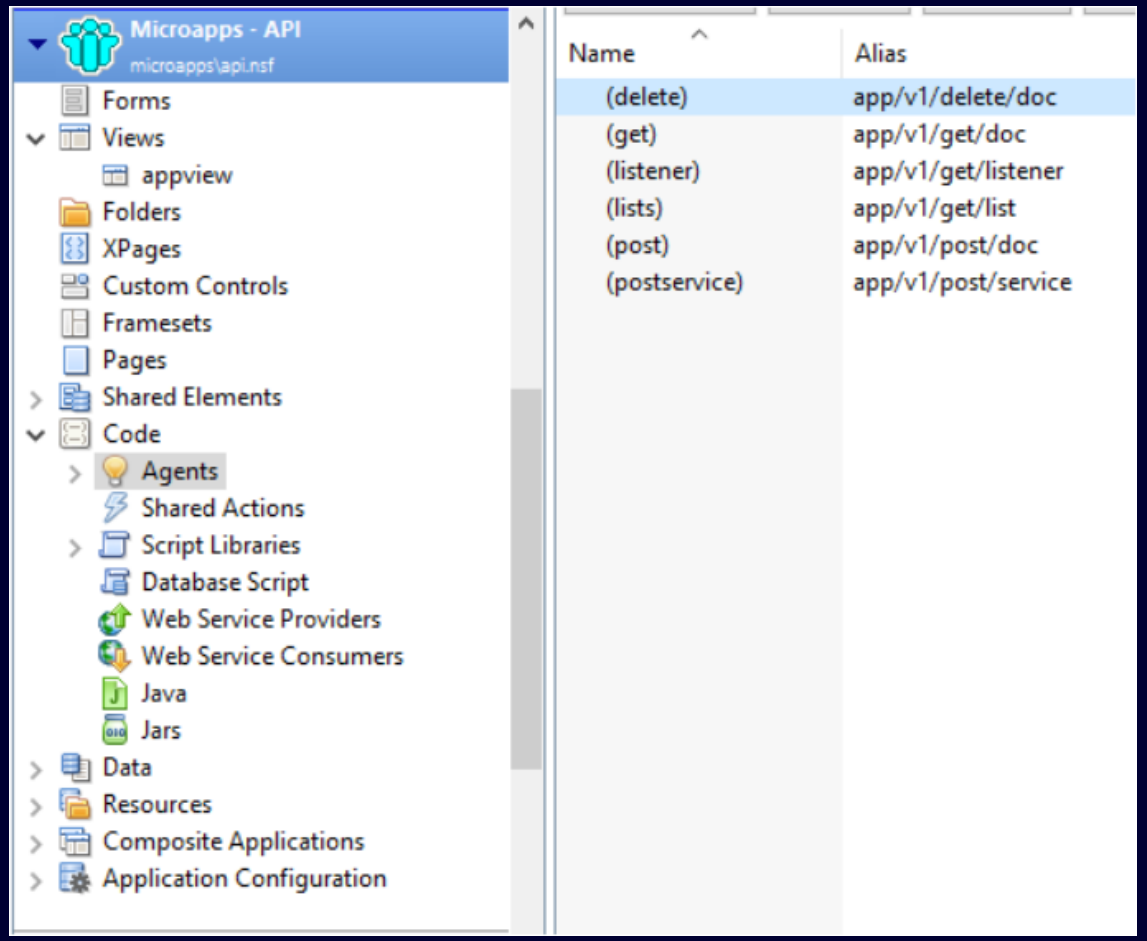
Domino Microapp Contains:

- Reusable framework that can be used by all apps
- Allow the creator to focus only on the UI experience
- Allow the creator the ability to create microapps using:
  - No-code
  - Low-code
  - Vibe-code

Domino-based Microapp Architecture

Fix supported APIs endpoints:

- app/v1/delete/doc
- app/v1/get/doc
- app/v1/get/list
- app/v1/get/listener
- app/v1/post/doc
- app/v1/post/service



Domino-based Microapp Architecture

Domino Microapp Contains:

- Sync microservices (Run On Server Agents)
- Async microservices (Scheduled Server Agents)
- Not true microservices, they do not replicate and scale when needed but provides the same concept

Domino-based Microapp Architecture

Event Bus Driven:

- Allow for Plug and play integration
- Integration to Web services
- Integration to Artificial Intelligences (AI)

Summary

- Domino provides all the tools for you to create an enterprise microapp platform
- Rethink what Domino databases are, as secure Web app containers
- Building microapps on Domino provide one key advantage over other platforms, security

Summary

Rethink and Reimagine

ENGAGE user group

Creating a Service Oriented Architecture (SOA) Platform with Domino

October 16, 2025

https://www.youtube.com/watch?v=tOnVthcR1Yg

Please support

https://domino.ideas.hcl-software.com/ideas/DOMINO-I-3189
Domino Support on ARM and Apple Silicon

https://domino.ideas.hcl-software.com/ideas/DOMINO-I-2285
ARM version of Domino Server on Linux

My Information

LinkedIn

- https://linkedin.com/taishanworks

OpenNTF Discord Channel

Blog

- https://dominointerface.blogspot.com