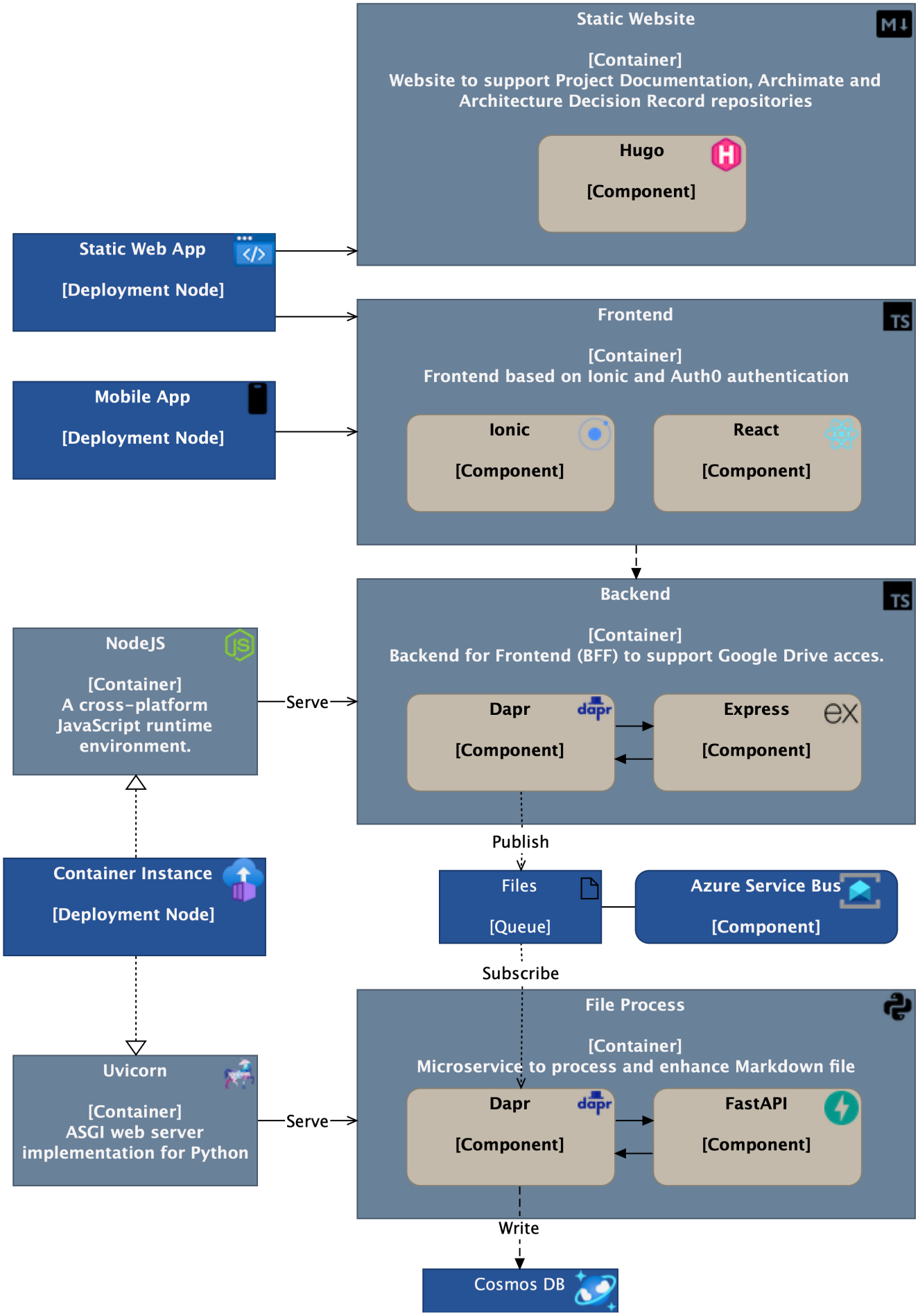
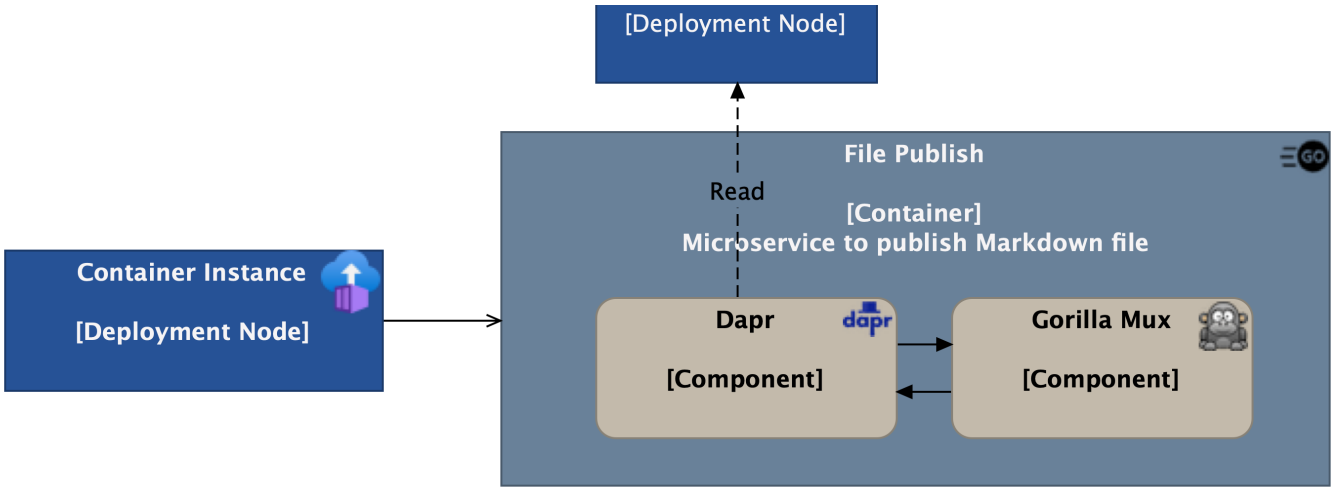
Deployement Diagram





Introduction

Static Web App hosted on deployment node. The architecture is composed of two main containers:

> - Frontend Container: Built using Ionic and React components. Provides a user interface that integrates with Auth0 for secure authentication. This container is TypeScript-based, ensuring modern, strongly-typed development.
> - Static Website Container: Built using Hugo for generating static content. Designed to host project documentation, Archimate diagrams, and Architecture Decision Records (ADRs).

Both containers are deployed under Static Web App deployment node, which serves as the hosting environment for delivering the application and documentation seamlessly. This architecture ensures modularity, security, and ease of deployment.

Container-based architecture integrating multiple components for a backend workflow.

> - NodeJS Container serves as a Backend-for-Frontend (BFF) to facilitate Google Drive access. It uses Express for API handling and Dapr (cf. ADR 0017-distributed-application)as a component to publish events/messages.
> - These events are sent to an Azure Service Bus queue named Files (cf. ADR 0018-add-publish-and-subscribe-components).
> - A File Processing Microservice implemented with FastAPI (Python) subscribes to the Azure Service Bus queue to process messages. The microservice runs in a Uvicorn container and leverages Dapr (cf. ADR 0017-distributed-application) for interaction.
> - The containers are deployed on a Container Instance, providing a scalable deployment environment for both NodeJS and Uvicorn services.

This architecture demonstrates a decoupled and event-driven system, where the Backend publishes file events, and the File Processing Microservice asynchronously processes the Markdown files. Dapr facilitates communication between the services.

NodeJS (Application Component)

Properties

Stereotype	URL
Container	https://nodejs.org/

Relationships

From	Relationship	To	Name/Label	Description
NodeJS	Serving Relationship	Backend (Application Component)	Serve	

A cross-platform JavaScript runtime environment.

Backend (Application Component)

Properties

Stereotype	Github
Container	https://github.com/mickael-royer/unicorn-express-server

Relationships

From	Relationship	To	Name/Label	Description
Backend	Assignment Relationship	Express (Application Function)		
Backend	Assignment Relationship	Dapr (Application Function)		

Backend for Frontend (BFF) to support Google Drive acces.

Express (Application Function)

Properties

Stereotype	URL
Component	https://expressjs.com

Relationships

From	Relationship	To	Name/Label	Description
Express	Triggering Relationship	Dapr (Application Function)		

Express is a fast, unopinionated, minimalist web framework for Node.js, providing a robust set of features for web and mobile applications.

Dapr (Application Function)

Properties

Stereotype	URL
Component	https://dapr.io

Relationships

From	Relationship	To	Name/Label	Description
Dapr	Access Relationship (write)	[Queue] Files (Artifact)	Publish	
Dapr	Triggering Relationship	Express (Application Function)		

Dapr (Distributed Application Runtime) is an open-source, portable, event-driven runtime that helps developers build microservices applications. It abstracts away the complexities of common microservices infrastructure tasks like service discovery, state management, message brokers, and pub/sub systems. Dapr provides building blocks for microservices, enabling developers to focus on writing business logic instead of dealing with infrastructure challenges.

Container Instance (Node)

Properties

Stereotype
Deployment Node

Relationships

From	Relationship	To	Name/Label	Description
Container Instance	Realization Relationship	NodeJS (Application Component)		
Container Instance	Realization Relationship	Uvicorn (Application Component)		

Azure Service Bus (Technology Service)

Azure Service Bus is a fully managed enterprise message broker service provided by Microsoft Azure. It supports various messaging patterns like queues, topics, and subscriptions, helping applications communicate asynchronously.

[Queue] Files (Artifact)

Relationships

From	Relationship	To	Name/Label	Description
[Queue] Files	Association Relationship	Azure Service Bus (Technology Service)		

Queues: Used for one-to-one communication, where a message is sent to a queue and a single consumer reads it.

Uvicorn (Application Component)

Properties

Stereotype	URL
Container	https://www.uvicorn.org/

Relationships

From	Relationship	To	Name/Label	Description
Uvicorn	Serving Relationship	File Process (Application Component)	Serve	

ASGI web server implementation for Python

File Process (Application Component)

Properties

Stereotype
Container

Relationships

From	Relationship	To	Name/Label	Description
File Process	Assignment Relationship	Dapr (Application Function)		
File Process	Assignment Relationship	FastAPI (Application Function)		

Microservice to process Markdown file

Dapr (Application Function) 2

Properties

Stereotype	URL
Component	https://dapr.io

Relationships

From	Relationship	To	Name/Label	Description
Dapr	Access Relationship (read)	[Queue] Files (Artifact)	Subscribe	
Dapr	Triggering Relationship	FastAPI (Application Function)		

Dapr (Distributed Application Runtime) is an open-source, portable, event-driven runtime that helps developers build microservices applications. It abstracts away the complexities of common microservices infrastructure tasks like service discovery, state management, message brokers, and pub/sub systems. Dapr provides building blocks for microservices, enabling developers to focus on writing business logic instead of dealing with infrastructure challenges.

FastAPI (Application Function)

Properties

Stereotype	URL
Component	https://fastapi.tiangolo.com/

Relationships

From	Relationship	To	Name/Label	Description
FastAPI	Triggering Relationship	Dapr (Application Function)		

FastAPI is a modern, fast (high-performance), web framework for building APIs with Python based on standard Python type hints.

Frontend (Application Component)

Properties

Stereotype	Github
Container	https://github.com/mickael-royer/unicorn-ionic

Relationships

From	Relationship	To	Name/Label	Description

From	Relationship	To	Name/Label	Description
Frontend	Assignment Relationship	Ionic (Application Function)		
Frontend	Assignment Relationship	React (Application Function)		

Frontend based on Ionic and Auth0 authentication

React (Application Function)

Properties

Stereotype
Component

Ionic (Application Function)

Properties

Stereotype
Component

Static Website (Application Component)

Properties

Stereotype	Github
Container	https://github.com/mickael-royer/unicorn-hugo-website

Relationships

From	Relationship	To	Name/Label	Description
Static Website	Assignment Relationship	Hugo (Application Function)		

Website to support Project Documentation, Archimate and Architecture Decision Record repositories

Hugo (Application Function)

Properties

Stereotype	URL
Component	https://gohugo.io/

Static Web App (Node)

Properties

Stereotype
Deployment Node

Relationships

From	Relationship	To	Name/Label	Description
Static Web App	Serving Relationship	Frontend (Application Component)		
Static Web App	Serving Relationship	Static Website (Application Component)		

Managed platform for building and deploying static web applications.