



universidade de aveiro
departamento de eletrónica,
telecomunicações e informática

Evolving Piranha CMS for Contents'R'Us

**Event-Driven Extensions & Secure
Integrations**

Software Architectures, Final Project

Daniel Madureira, 107603
João Andrade, 107969
José Gameiro, 108840
Tomás Victal, 109018



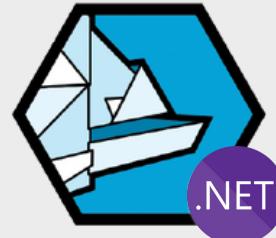
01

Overview

System Analysis

Contents'R'Us

CMS system derivated from Piranha CMS



Piranha CMS

Lightweight, cross-platform content management system built using .NET Core



Client Demands shifted!

Asynchronous platform

Support for inbound and outbound data

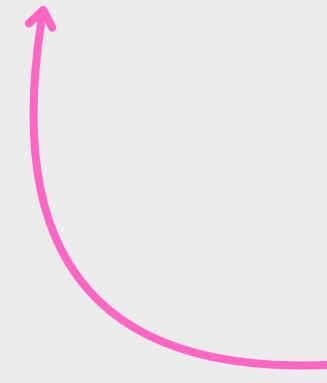
Secure integration with external services

Our Scenario

Transform the CMS into a modern, **asynchronous** platform that supports inbound and outbound data flows, integrates easily with external services and leverages secure messaging

System Analysis Disparities!!

Synchronous Design Bias



Asynchronous platform

Support for inbound and outbound data

Secure integration with external services

Event driven architecture

Security Limitations for Integrations

No Inbound Event Handling



Lack of Event Infrastructure



Project Vision

Our main vision for this assignment is to transform Piranha CMS, into a **modern, asynchronous content platform** that is integration-ready and resilient.

■ **Loose Coupling**

■ **Secure Communication**

■ **Configurability**

Allowing external systems to push and pull data dynamically, while maintaining:

■ **Trust**

■ **Integrity**

■ **Performance**

Custom Publish/Subscribe Mechanism

Inbound Integration via External Publishers

Secure Configurable Messaging Infrastructure



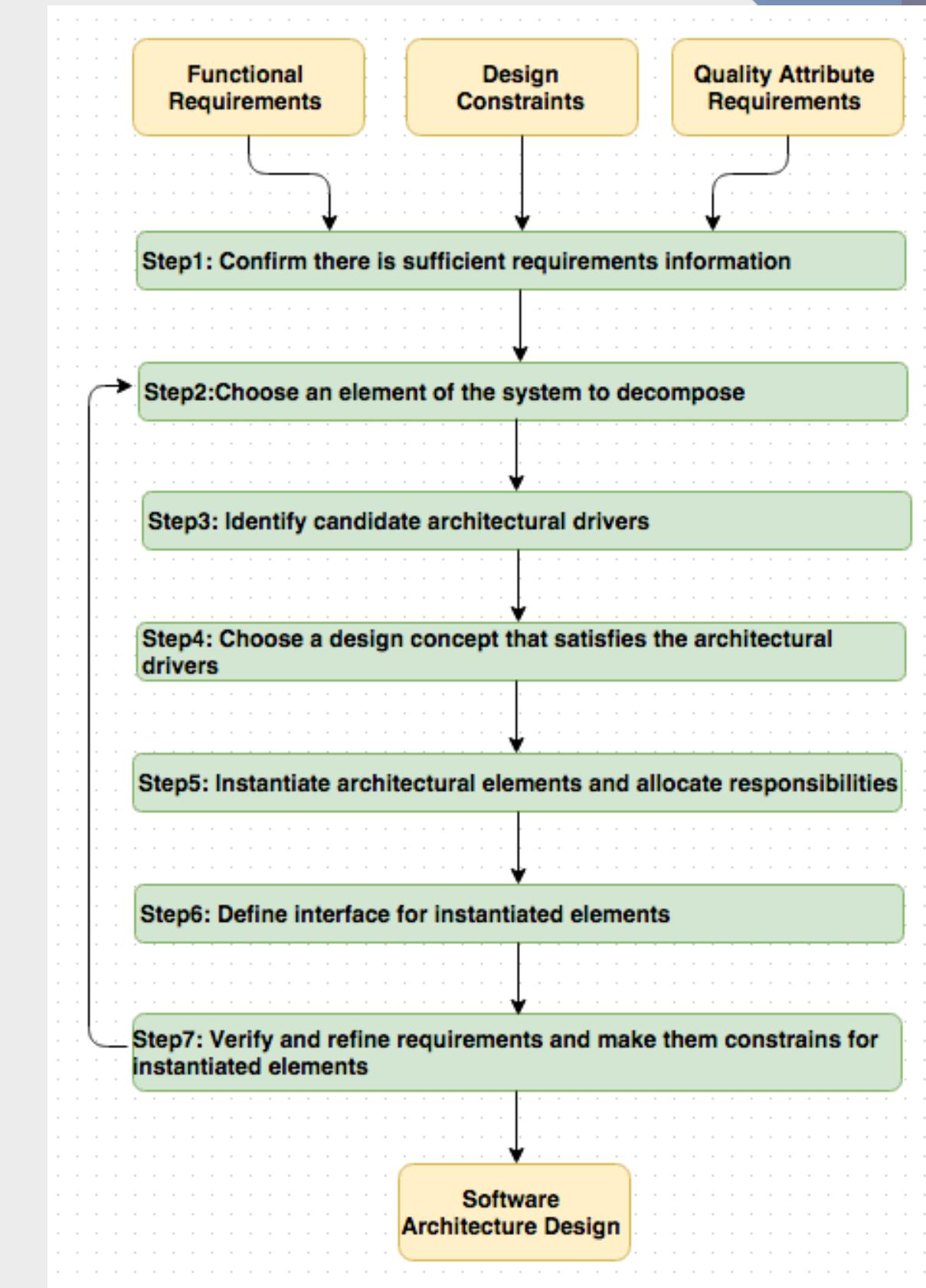
02

Architectural Design Methodology

ADD

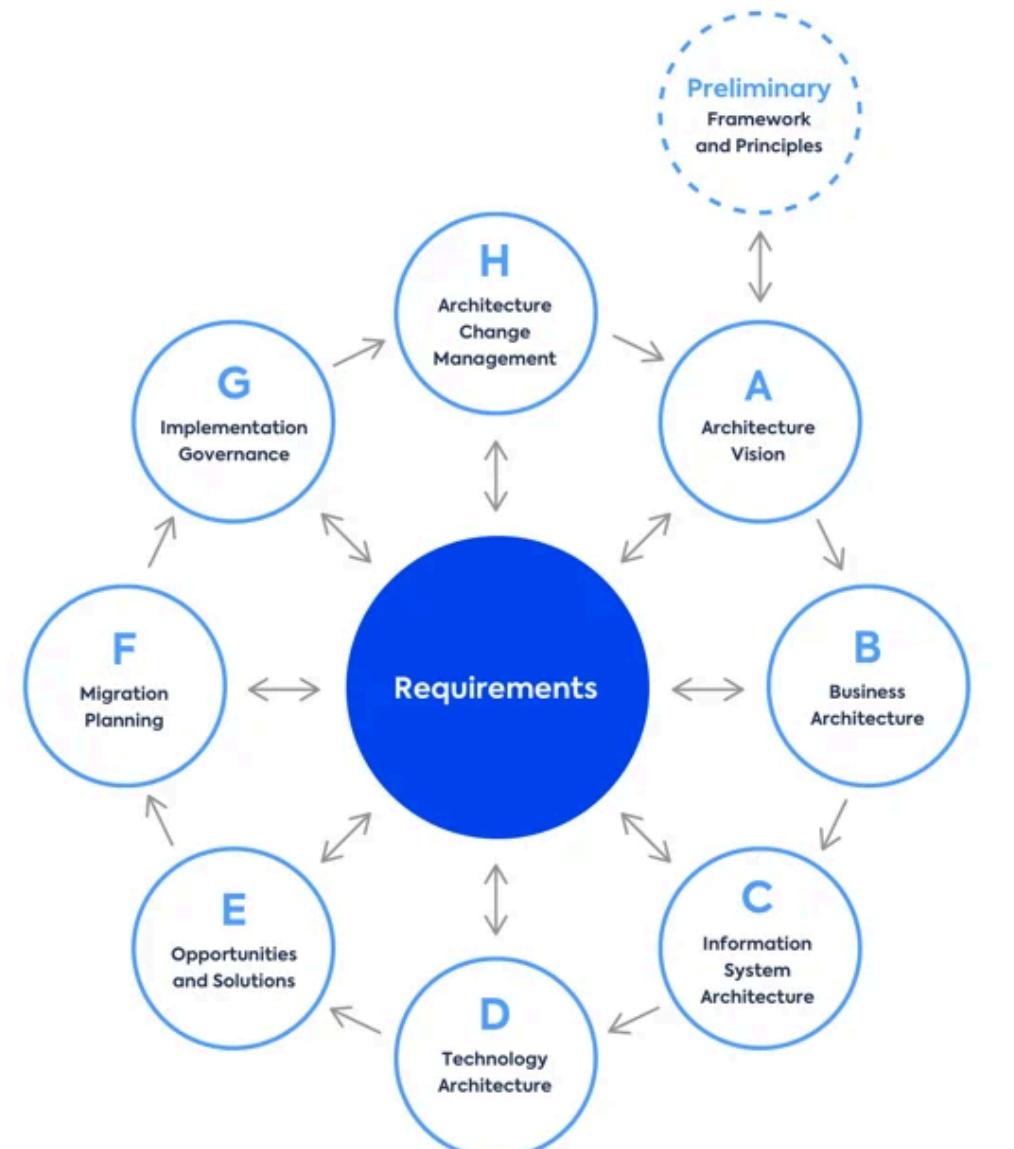
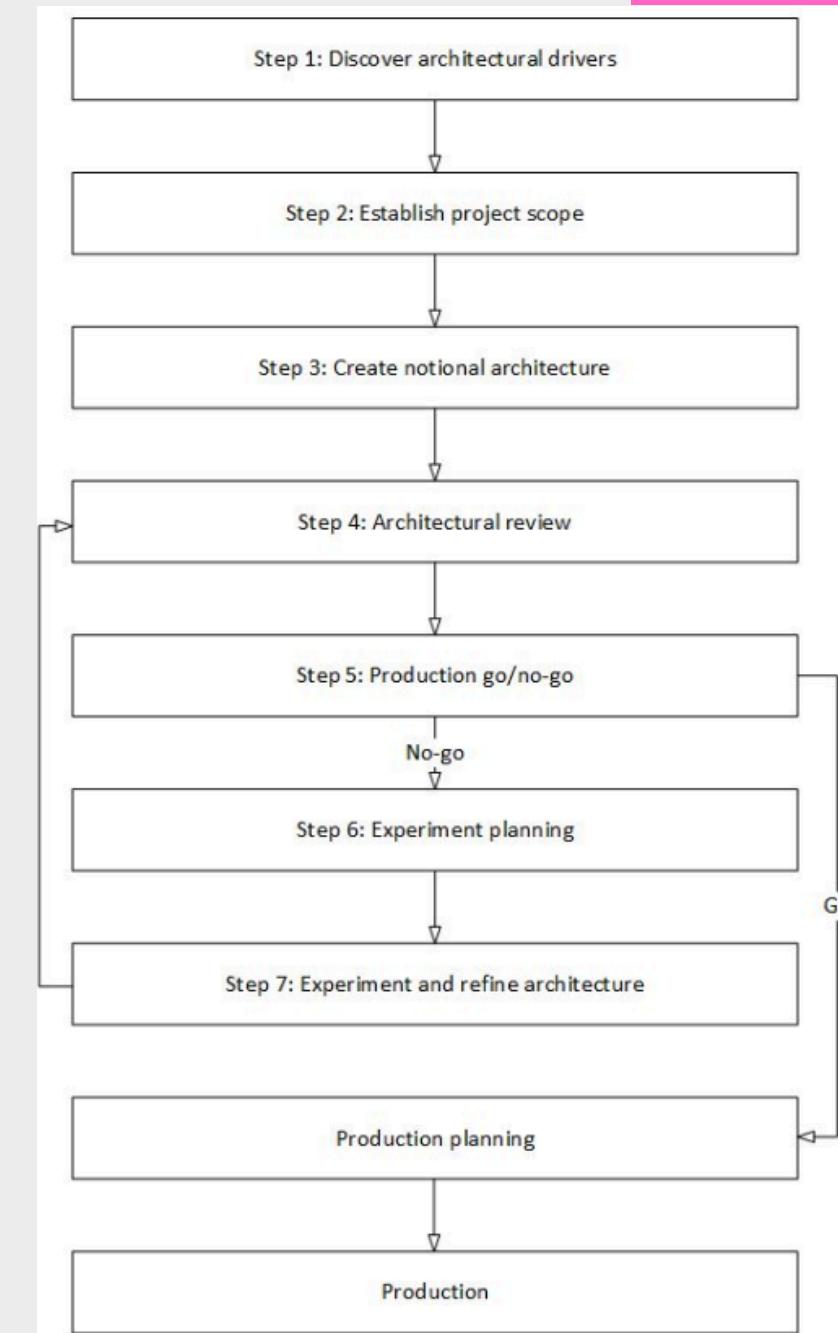
Attribute-Driven Design

- Focuses on **quality attributes**
 - **Top-down** and **iterative** approach
 - **Architectural tactics**
 - Complements **Domain-Driven Design** principle



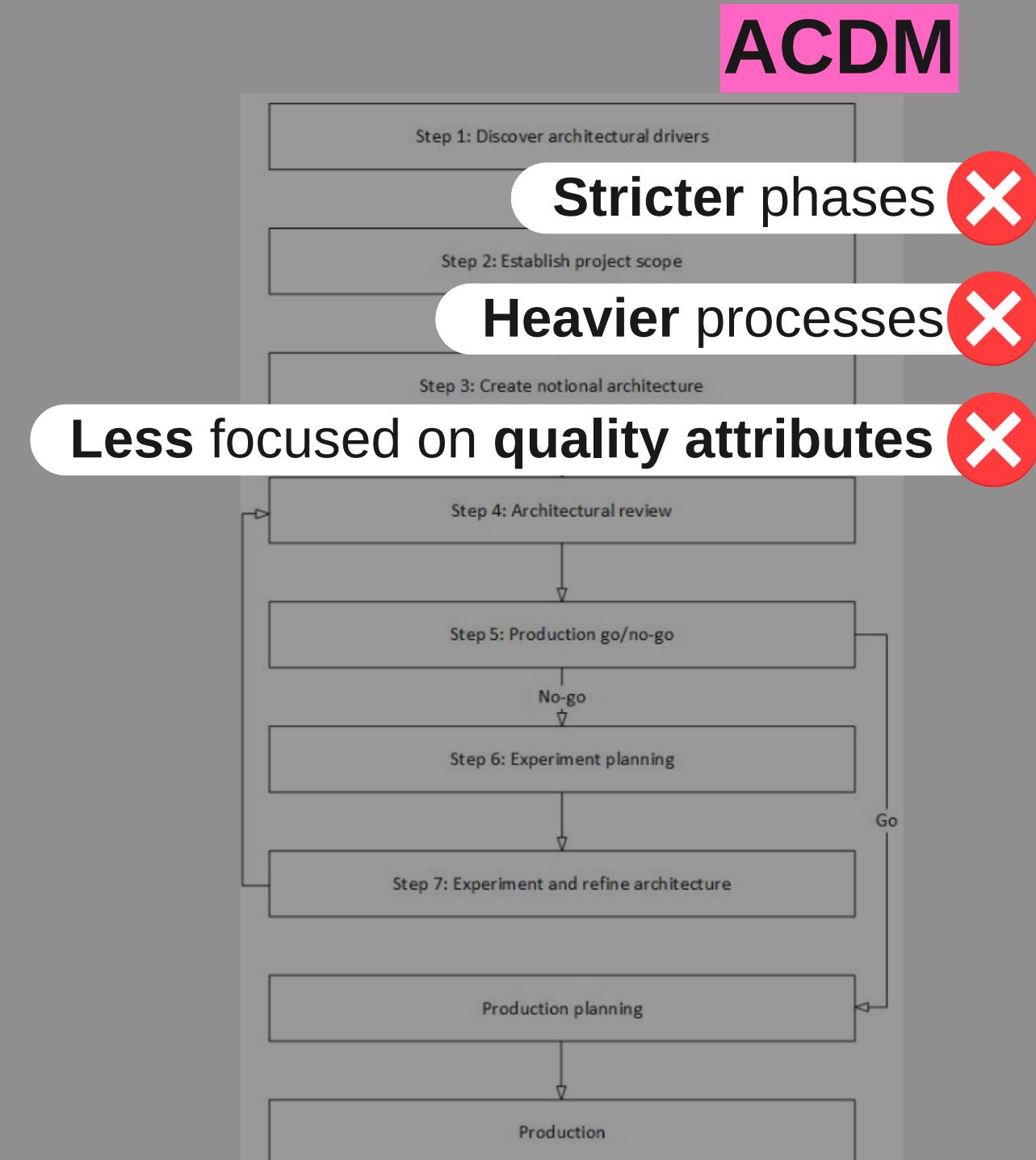
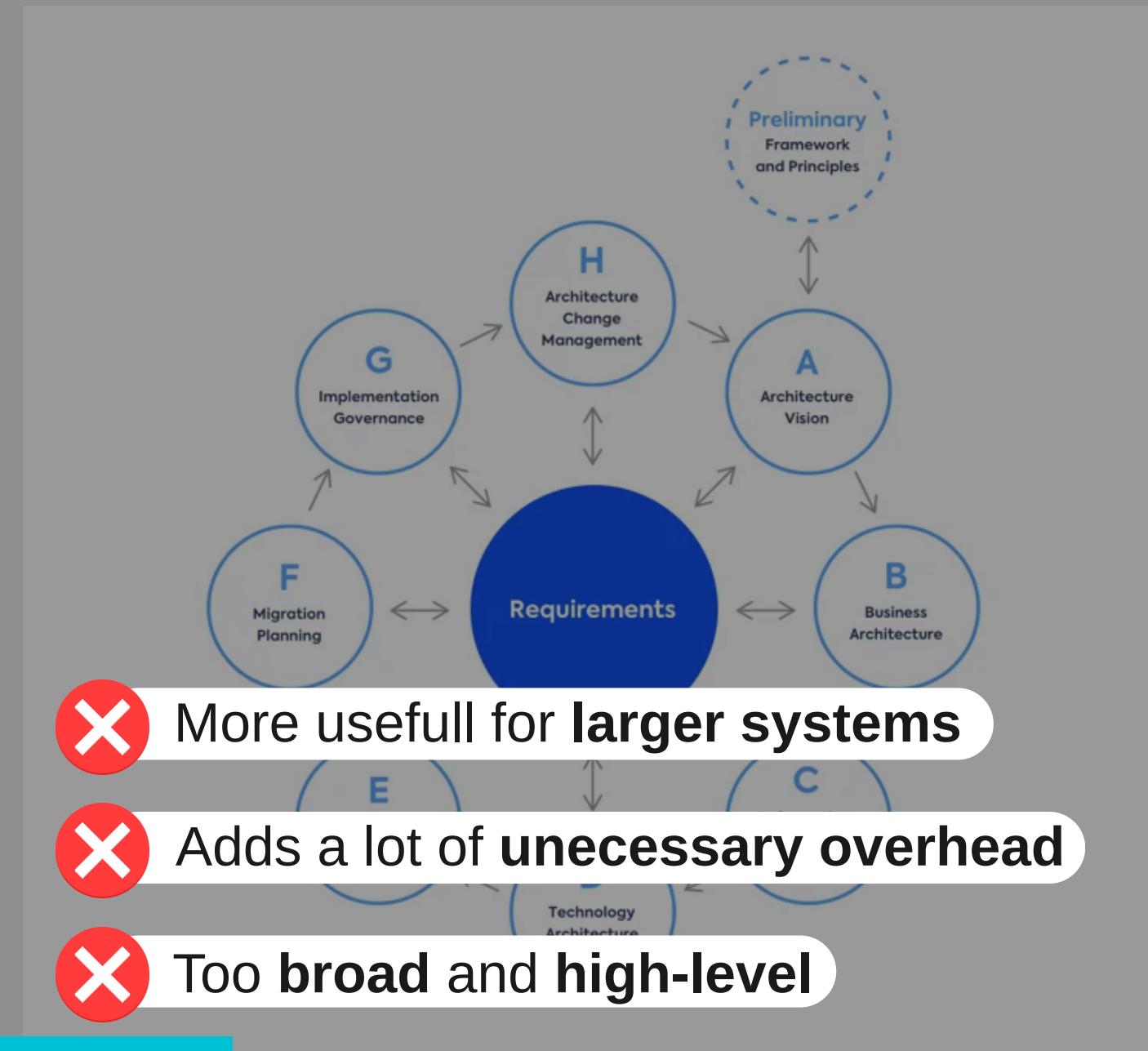
Alternatives

ACDM



TOGAF

Alternatives



03 Requirements

Functional Requirements

ID	Requirement	Actor
FR-1	Define possible actions for external publishers	Admin
FR-2	Publish new content	External Publisher
FR-3	See all the existing events	Admin
FR-4	Decide if a given content will be published	Admin
FR-5	Subscribe and unsubscribe to existing events	External Publisher
FR-6	Define authorized external publishers	Admin
FR-7	See new published events	Admin External Publisher
FR-8	Needs to integrate easily with external services	Admin External publisher

Non-Functional Requirements

ID	Requirements	Category
NFR-1	Event processing (publishing and inbound) must occur with a maximum latency of 1 second under normal load.	Performance
NFR-2	The system must support up to 10,000 concurrent event subscriptions and 1,000 inbound events per minute without service degradation.	Scalability
NFR-3	All event communications (inbound and outbound) must be authenticated and signed to ensure integrity and authenticity.	Security
NFR-4	The system should achieve 99% uptime for the Event Manager and Inbound Handler modules.	Availability
NFR-5	Code for new modules (Event Manager, Inbound Handler) should follow clean architecture principles and allow easy extension.	Maintainability
NFR-6	Failed event deliveries must be automatically retried up to 3 times with exponential backoff, before moving to a dead letter queue.	Reliability
NFR-7	Admin users must be able to configure events, endpoints, and keys without modifying application code.	Configurability
NFR-8	The admin configuration interface must allow an event to be fully configured (published or subscribed) within 5 clicks.	Usability

Constraints

ID	Constraint
CON-1	You must base your work on the existing Piranha CMS (.NET Core, C#) framework.
CON-2	Must use open-source or .NET-compatible libraries (no paid 3rd-party tools without prior approval).

04 Domain Driven Design

Core Domains



Event Publishing Domain

Responsible for defining, producing, and distributing **internal domain events**.



Subscription Management Domain

Allows external systems to show interest in specific events and what type of events they wish to receive.



Security and Access Control Domain

Manages access to the publishing and integration infrastructure.



External Integration and Inbound Event Handling Domain

Enables secure, structured receipt of events from trusted external publishers.

Supporting Subdomains

Monitoring & Observability

Tracks delivery stats, failed attempts, retries, and logs audit trails for compliance.



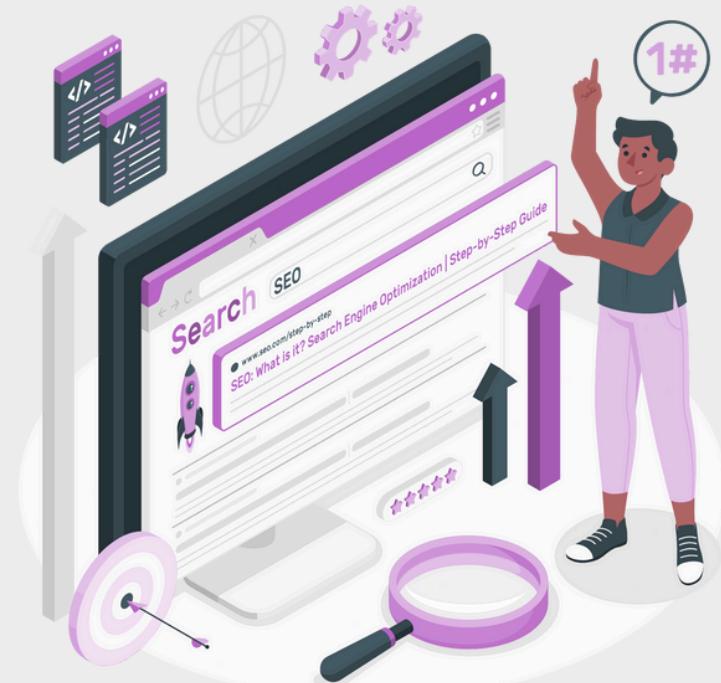
Generic Subdomains

User Management

Utilizes Piranha CMS's built-in identity features for managing admin-level users.

CMS Platform

Use Piranha's framework as the base of the project for the content management system.



05

Cross-Cutting Concerns

Cross-Cutting Concerns

Security

Use token-based authentication, api keys and message signing to ensure secure and tamper-proof communication.

Logging

The system should log events to make it easier to debug problems and to understand what has happened within the system.

Resilience

Make sure the system doesn't lose the messages that are supposed to be sent and stores them if the external system is offline.

Observability

Health checks, real-time metrics, and alerts will be used to provide full visibility into event flow and system status.

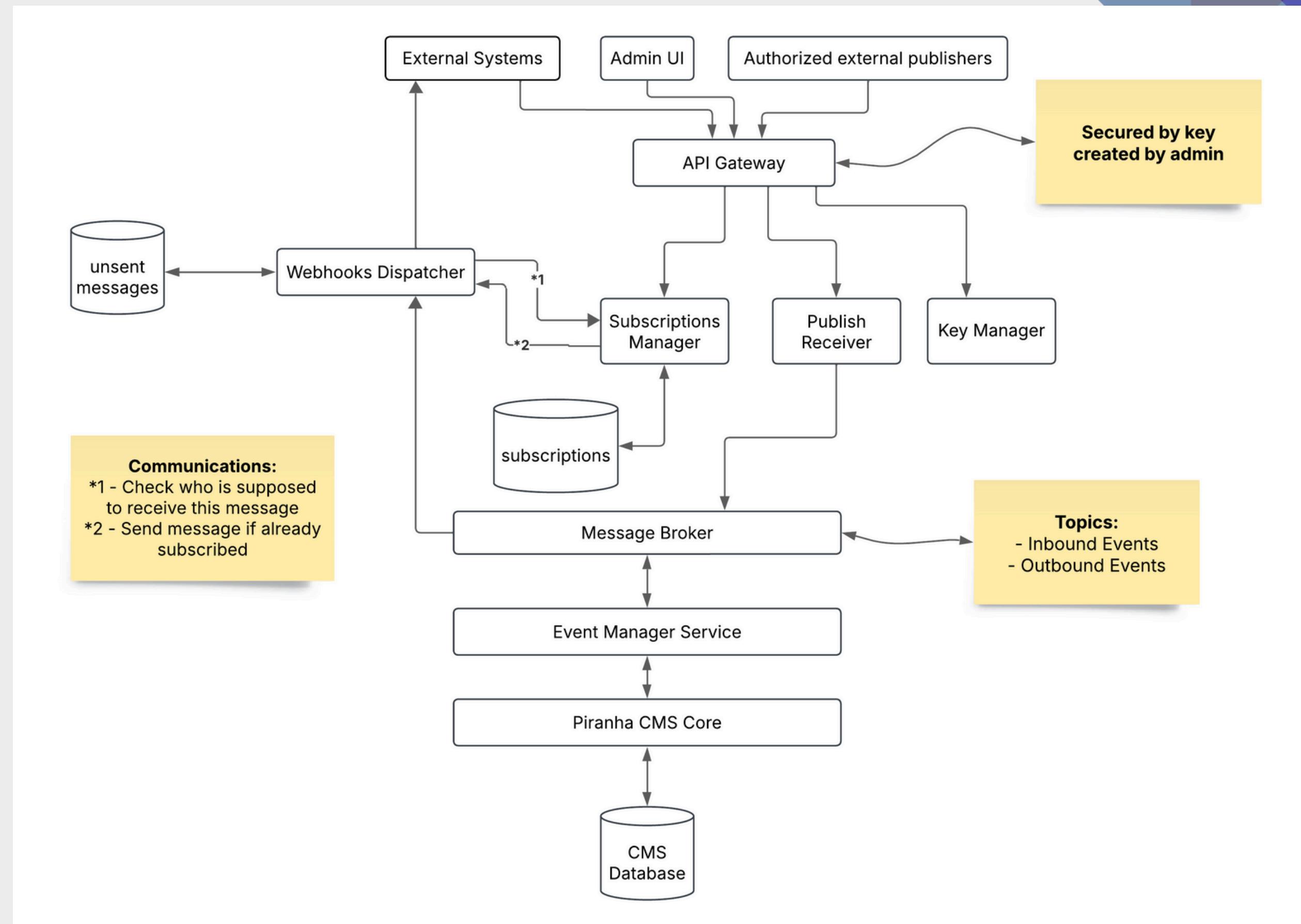
Scalability

The system needs to be able to scale as demands grow, such as when serving more clients. The asynchronous nature of the message queue makes this possible.

06

Proposed Architecture & Roadmap

Envisioned Architecture



Road Map



Event Publishing Infrastructure

Build Event Manager Service and integrate with Message Broker to capture internal events and support publishing from authorized clients.



Webhook Delivery System

Develop Webhooks Dispatcher to send event payloads to subscribers based on registered subscriptions.



Subscription Management Module

Implement Subscriptions Manager to handle registration, updates, deletions, and event filtering for targeted delivery.

Thanks

Does anyone have
questions?

