Piranha CMS Multi-Tenanc

Miguel Figueiredo, Miguel Cruzeiro, Diogo Silva, Guilherme Amorim

Universidade de Aveiro, DETI, Software Architectures

03/06/2025

Agenda

01 Introduction

02 Project Management

03 User Stories

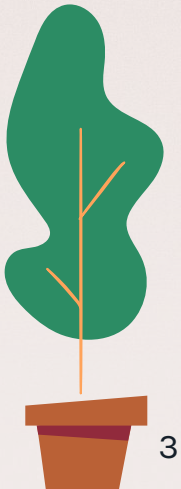
04 Live Demo

05 Conclusions

01

What is Multi-Tenancy?

- “Any environment that onboards, deploys, manages, and operates tenants through a single, unified experience.”
- A Shared infrastructure with isolated tenant experiences
- Each tenant (customer) expects control and isolation over their data, users, and configurations.



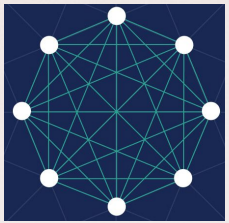
Project Vision

Strategic Objectives:

- Achieve strict tenant isolation (data, users, media)
- Enable automated tenant provisioning (onboarding)
- Ensure horizontal scalability through modular or microservices architecture
- Ensure observability with per-tenant resource metering

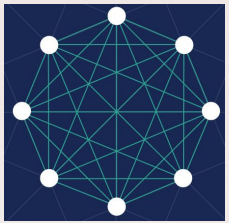
Key Quality Attributes:

- Security
- Isolation
- Scalability
- Performance
- Onboarding Usability
- Efficiency

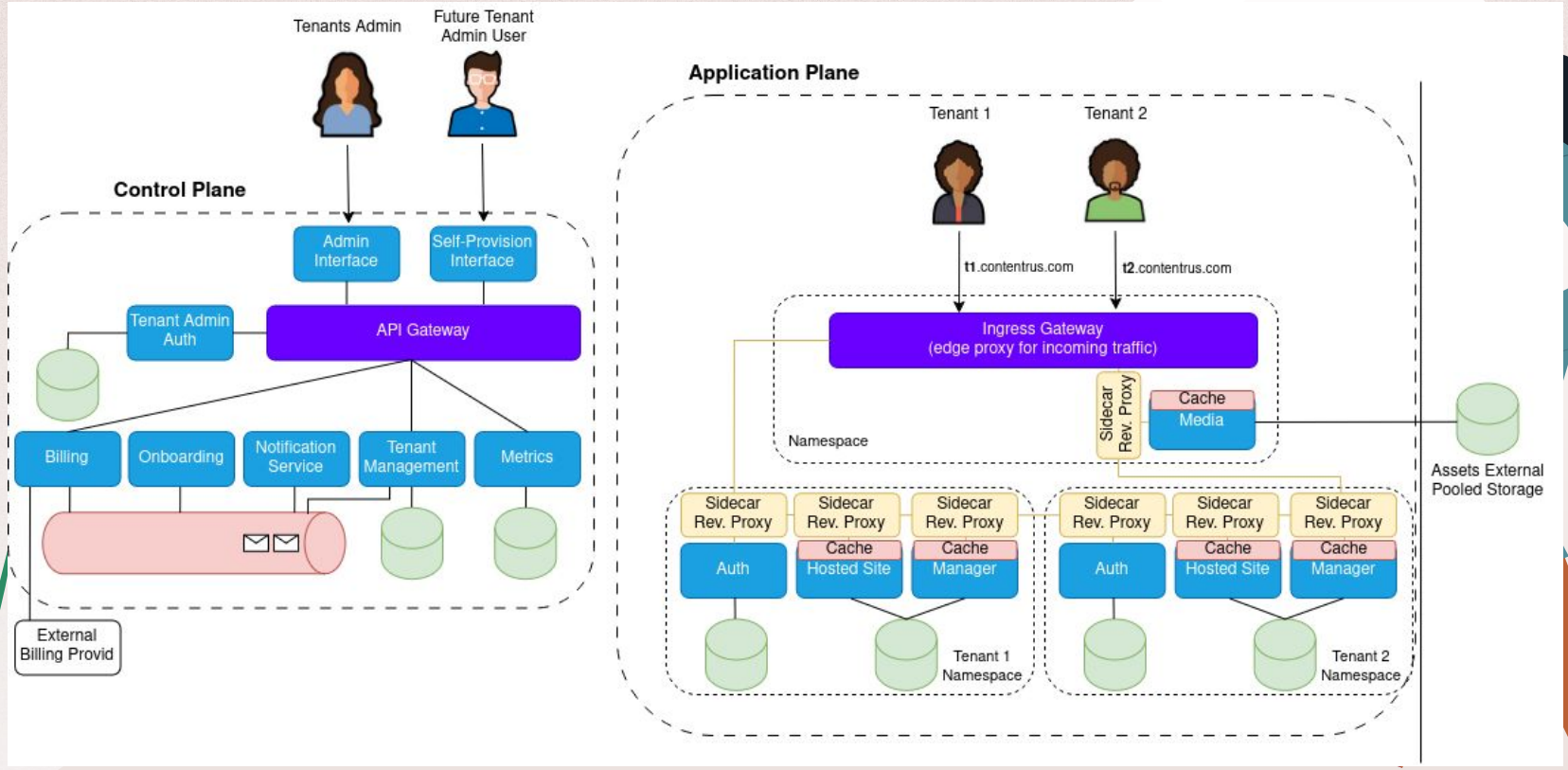


Piranha CMS Limitations

- **User Management** – global user management with no concept of tenancy
- **Tenant Management & Onboarding** – as there is no tenant concept, there is no functionality for automated provisioning and onboarding
- **Data Isolation** – true multi-tenancy requires strict guarantees that one tenant cannot access another's data
- **Observability with Resource Management** – doesn't provide mechanisms to meter manage or meter resource consumption on a site, user or per-tenant basis
- **Monolithic Architecture & Scalability Concerns** – monolithic architecture limits scaling, as it would involve replicating the whole application instance because of a specific functionality (overprovisioning)

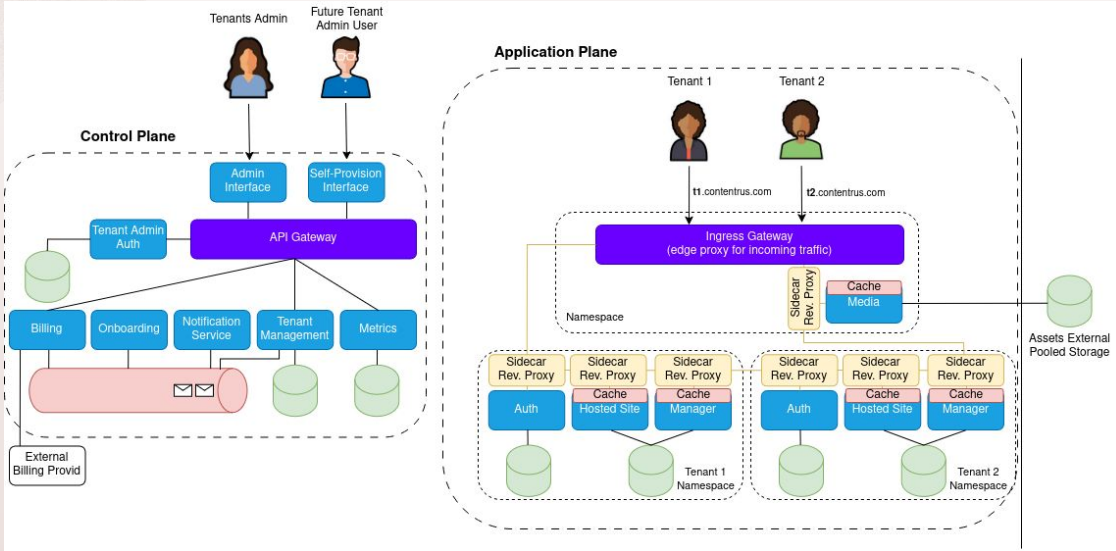


Piranha CMS - Proposed Architecture

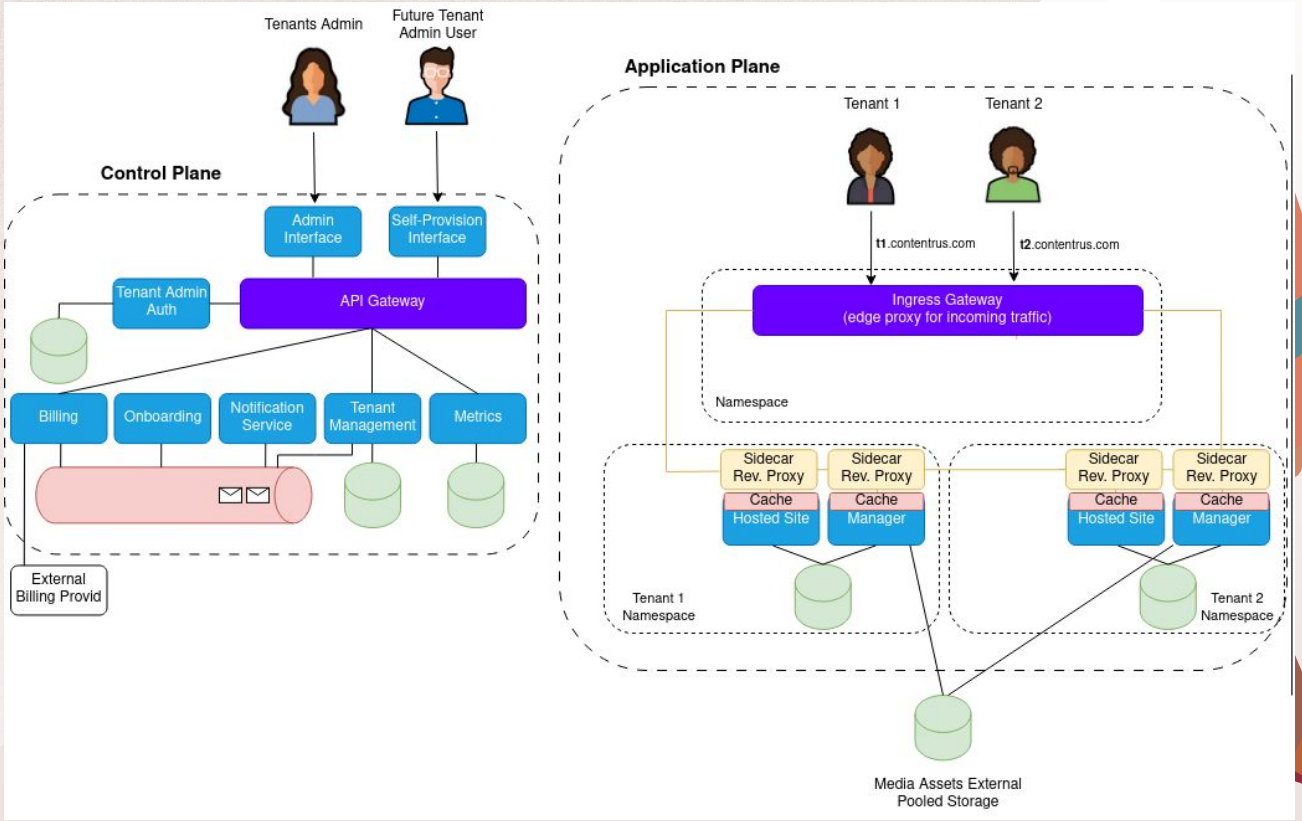


Piranha CMS - Proposed Architecture

- Control & Application Planes for tenant provisioning and per-tenant functionalities, respectively.
- Segregation of Original PiranhaCMS in Microservices
- Shared media library
- Control Plane includes API Gateway and Message Queue, for abstracting multiple microservices and promote event communication, respectively

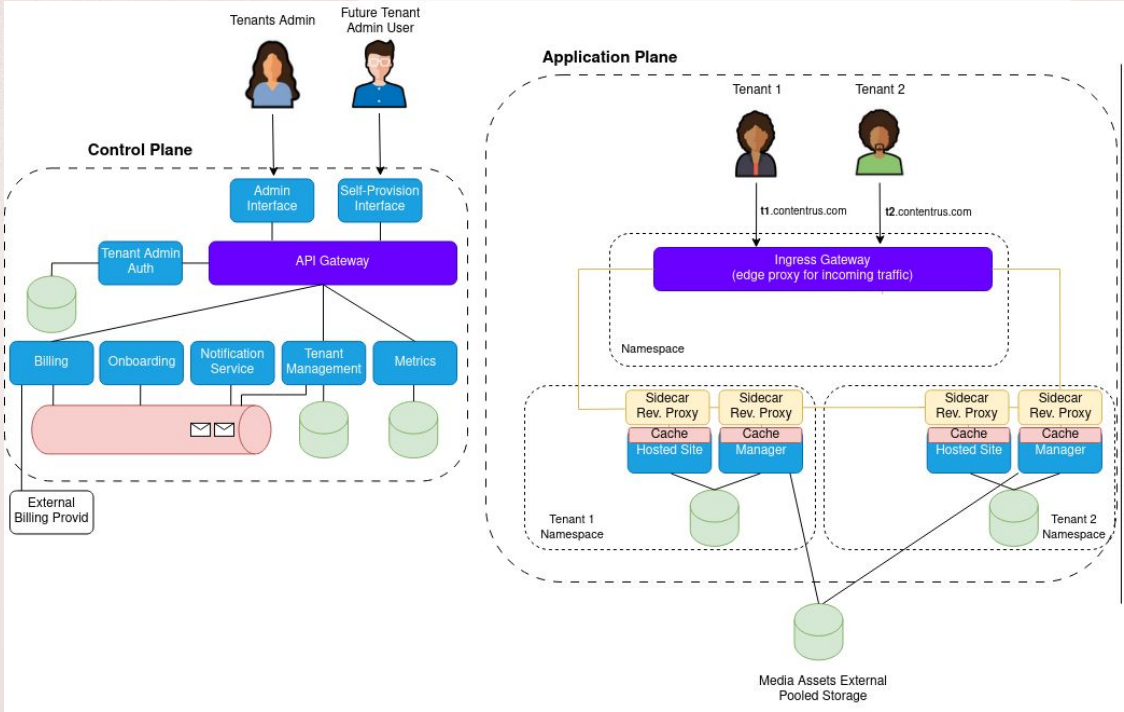


Piranha CMS - Actual Architecture



Piranha CMS - Actual Architecture

On the Application Plane the number of microservices was tuned down due to high coupling in the Auth and Media packages of Piranha CMS



02

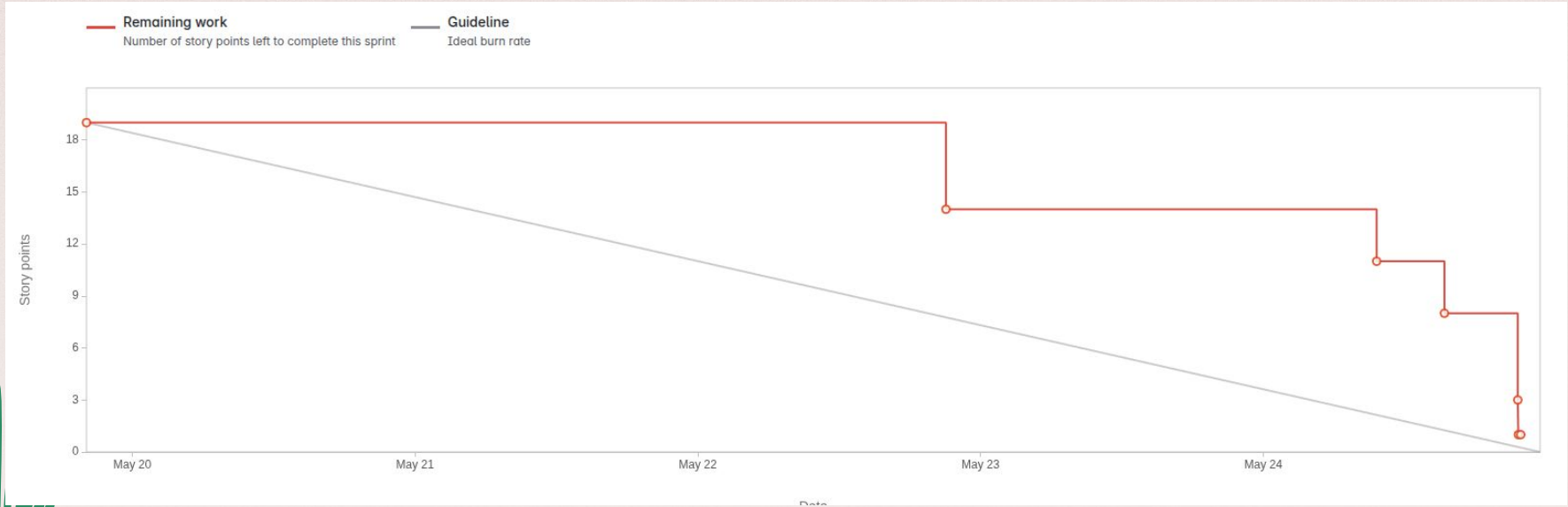
Project Management & Methodology

AGILE/SCRUM

- 4 Epics
  - Control Plane Base Infrastructure
  - Control Plane Core Services
  - Application Plane Base Infrastructure
  - CD Provisioning
- 8 User Stories
  - 4 Done (19 Story Points)
  - 1 In Progress (3 Story Points)
- 8 Tasks

Burndown Charts & Issues

Sprint 1



Burndown Charts & Issues

Sprint 1

Incomplete work items

View in issue navigator

Key	Summary	Issue type	Epic	Status	Assignee	Story points
SCRUM-2	Tenant Login	História	CONTROL PLANE CORE SE...	CONCLUÍDO	GA	1

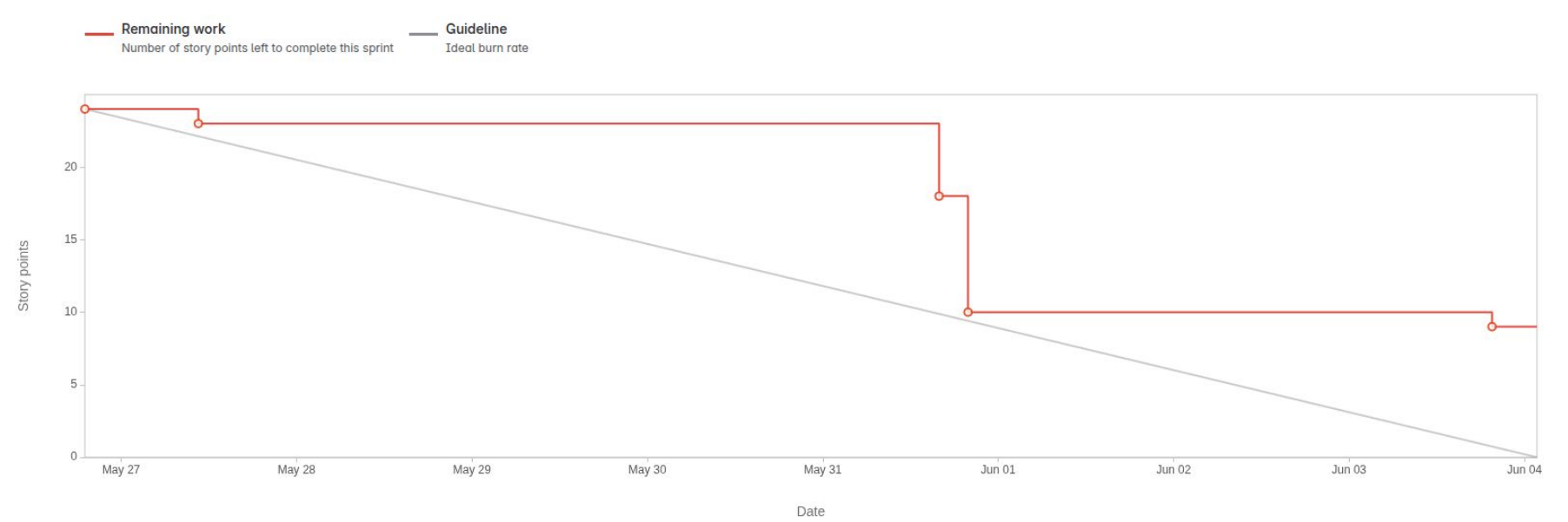
Completed work items

View in issue navigator

Key	Summary	Issue type	Epic	Status	Assignee	Story points
SCRUM-1	Tenant Registration	História	CONTROL PLANE CORE SE...	CONCLUÍDO	GA	3
SCRUM-11	Register Tenant Information (After Sign Up)	História	CONTROL PLANE CORE SE...	CONCLUÍDO	GA	2
SCRUM-12	Pick SaaS Subscription Tier & Billing	História	CONTROL PLANE BASE INF...	CONCLUÍDO	DS	5
SCRUM-7	Segregate Piranha CMS into HostedSite & Manager	Tarefa	APPLICATION PLANE BAS...	CONCLUÍDO	MF	3
SCRUM-13	Configure Istio Ingress Gateway and service mesh	Tarefa	APPLICATION PLANE BAS...	CONCLUÍDO	MF	5

Burndown Charts & Issues

Sprint 2



Burndown Charts & Issues

Sprint 2

Incomplete work items

View in issue navigator

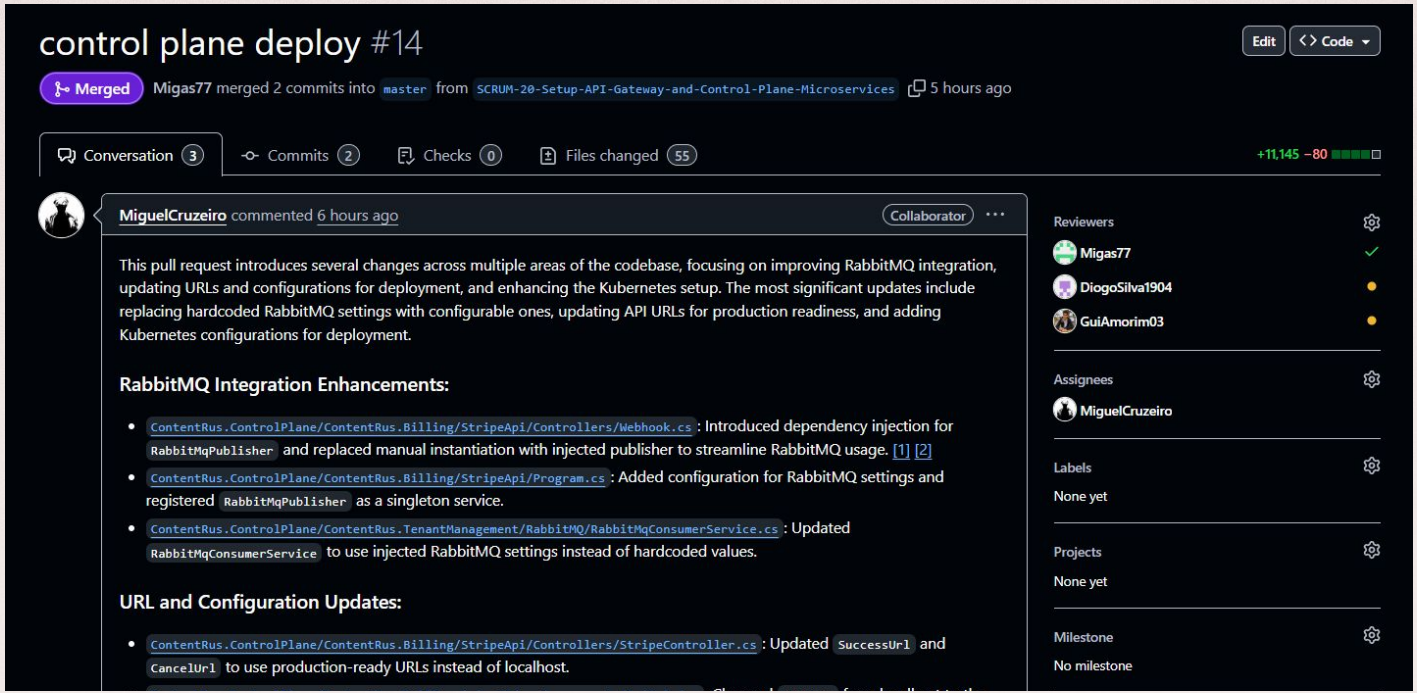
Key	Summary	Issue type	Epic	Status	Assignee	Story points
SCRUM-21	Observability Control Plane	Tarefa	CONTROL PLANE CORE SE...	A FAZER	M	3
SCRUM-23	Network and Authorization Policies for Isolation Between Tenants Namespaces	Tarefa	APPLICATION PLANE BAS...	A FAZER		3
SCRUM-25	Media Upload via Media Service	História	CONTROL PLANE CORE SE...	EM ANDAMENTO	GA	3

Completed work items

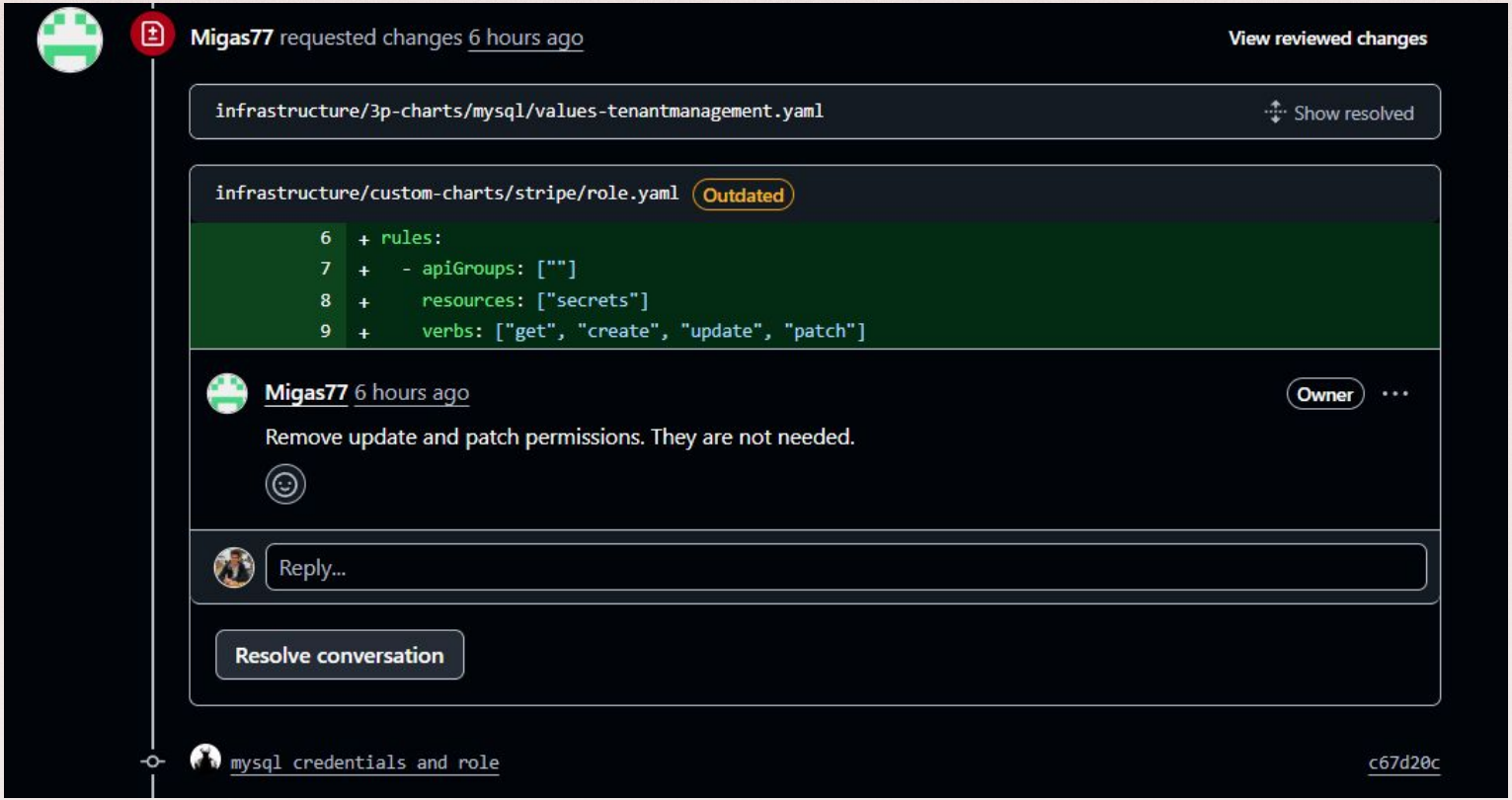
View in issue navigator

Key	Summary	Issue type	Epic	Status	Assignee	Story points
SCRUM-2	Tenant Login	História	CONTROL PLANE CORE SE...	CONCLUÍDO	GA	1
SCRUM-17	Tenant Onboarding & Environment Provisioning	História	CONTROL PLANE CORE SE...	CONCLUÍDO	MF	8
SCRUM-20	Setup API Gateway and Control Plane Microservices	Tarefa	CONTROL PLANE BASE INF...	CONCLUÍDO		1
SCRUM-22	Observability Application Plane On a Per-Tenant Basis	Tarefa	APPLICATION PLANE BAS...	CONCLUÍDO	M	5

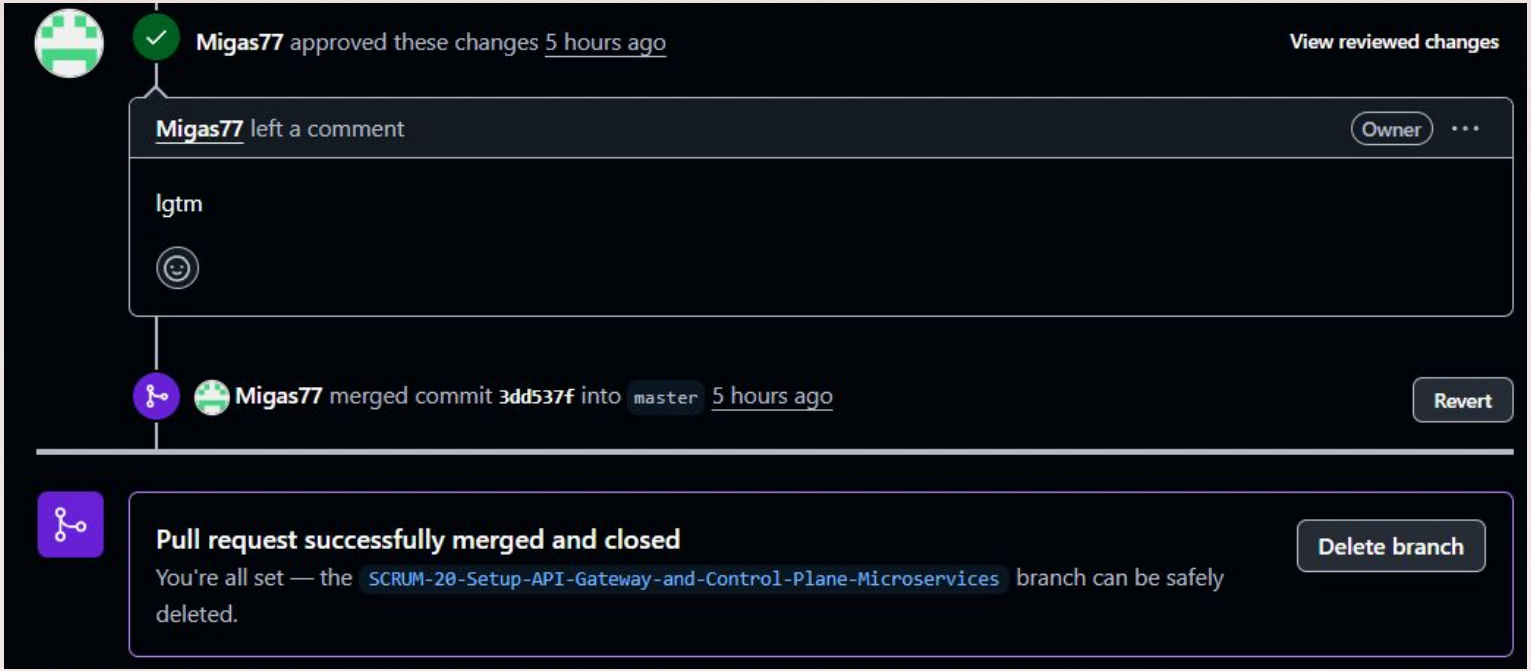
Pull Request



Pull Request



Pull Request



Devops

GitOps ensures that all tenant environments and platform configurations are stored in repositories

Argo Workflows orchestrates the creation of credentials and secrets. ArgoCD monitors Git to deploy the tenant's environment

Isolation

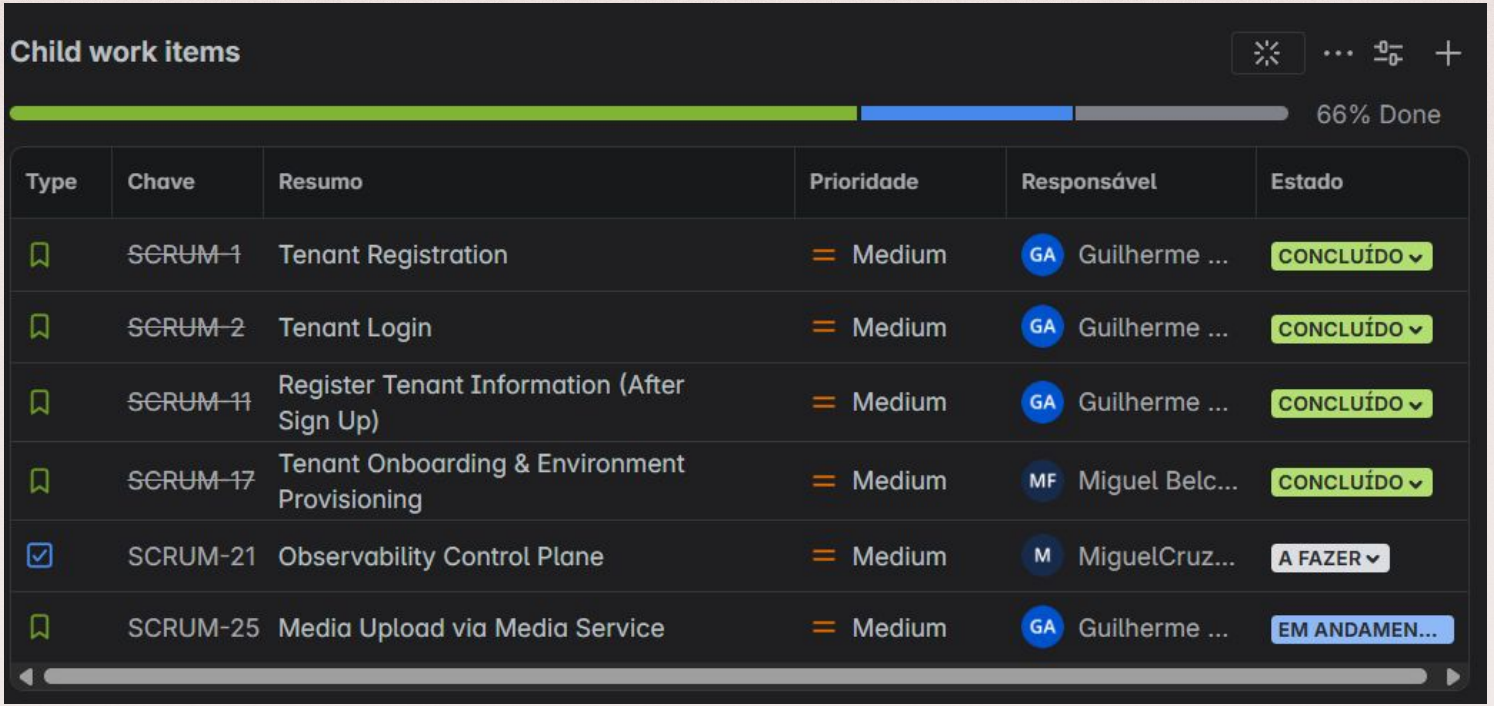
Network Policies block unauthorized traffic between tenant namespaces. This prevents data leaks and lateral movement across tenant boundaries in the Kubernetes cluster.

Authorization Policies control who can access tenant resources, ensuring that only authorized users or services can interact with a given tenant's environment.

03

User Stories

EPIC - Control Plane Core Services



Child work items

66% Done

Type	Chave	Resumo	Prioridade	Responsável	Estado
	~~SCRUM-1~~	Tenant Registration	= Medium	GA Guilherme ...	CONCLUÍDO
	~~SCRUM-2~~	Tenant Login	= Medium	GA Guilherme ...	CONCLUÍDO
	~~SCRUM-11~~	Register Tenant Information (After Sign Up)	= Medium	GA Guilherme ...	CONCLUÍDO
	~~SCRUM-17~~	Tenant Onboarding & Environment Provisioning	= Medium	MF Miguel Belc...	CONCLUÍDO
	SCRUM-21	Observability Control Plane	= Medium	M MiguelCruz...	A FAZER
	SCRUM-25	Media Upload via Media Service	= Medium	GA Guilherme ...	EM ANDAMEN...

STORY - Register Tenant Information (After Sign Up)

As a newly signed-up tenant,

I want to enter my tenant-specific information such as company name, address, and other relevant details,

So that my tenant profile is fully set up and ready for use.

STORY - Register Tenant Information (After Sign Up)

Acceptance Criteria

Given a registered tenant,

When they are redirected to the tenant information submission page after sign-up,

Then they should be able to enter information for fields company name, address

Given the user fills out the tenant information form,

When the user submits valid tenant information,

Then the tenant information is saved and linked to the user's account, and the user should be redirected to a page to pick its tier for the SaaS application

04

Live Demo

05

Conclusions