RailwayCabal

Initial Project Description

Team 12

Alex Bohlken
Abraham Dick
Noah Kenn
Qualen Pollard
Ryan Rodriguez

PROJECT DETAILS

Synopsis

Microservice based web-server framework designed to be scalable, secure, extensible and easily deployable.

Description

Advancements in container-based technologies, such as Docker, combined with the rise of cheap and accessible cloud-based services, such as Amazon Web Services and Microsoft Azure, has led the ability for developers to easily deploy web-based content distribution services across many server instances. This is in contrast to traditional web-based technologies that have relied on concurrency patterns for increasing performance. These older technologies rely on a monolithic based architecture, leading to large structures that make refactoring or modifying features difficult. Microservice architecture attempts to overcome this by segmenting each responsibility of a program into its own service and setting up a communication mechanism between them. The result is a composite structure that can vary in language and even platform between different responsibilities. Discrete services allow systems to easily grow and be pruned as products and business services change and grow without large refactoring efforts.

Unfortunately, long term scalability concerns are often beat out by costlier upfront development and the relative simplicity and availability of monolithic frameworks such as ExpressJS, Rails, and Django. RailwayCabal will provide the same ease of access as these frameworks while providing all benefits of the underlying microservice based architecture.

PROJECT LOGISTICS

Milestones

MILESTONE	ESTIMATION
Initial API, CORE Feature definitions	Nov 2018
Use Case Diagrams, CORE Coms Definition	Late Nov 2018
Static Content Delivery, process mgmt model, Second Channel Coms Impl'	Dec 2018
Process Mgmt Impl', First Channel Coms, Initial docs	Late Jan 2019
CORE services completeness, Integration testing and docs, Impl' plugin extension support	March 2019
Feature Complete, begin debugging, robustness testing, proofing	Early April 2019
Packaging, Documentation, Delivery	May 2019

Work Plan

CONCERN	ASSIGNMENT
Public facing routing and sub-routing, static content delivery	Alex B
First Channel Communications, IPC, Containerization	Abe D
Second Channel Communications, Statistics tracking	Noah K
CORE implementation, process permissioning	Qualen P
Code Generation, System Logistics	Ryan R

Budget

While deployment of RailwayCabal software might cost money via Amazon AWS or Microsoft Azure compute time, there will be no development cost and the completed software will be 100% operable without cost.