Computer Automation

Siddharth Garimella

Learning Goals

1. Building distributed systems
2. Integrating JS projects with different backends
3. Building user requirements into an MVP

Architecture

Studio

Build and test computer automations locally

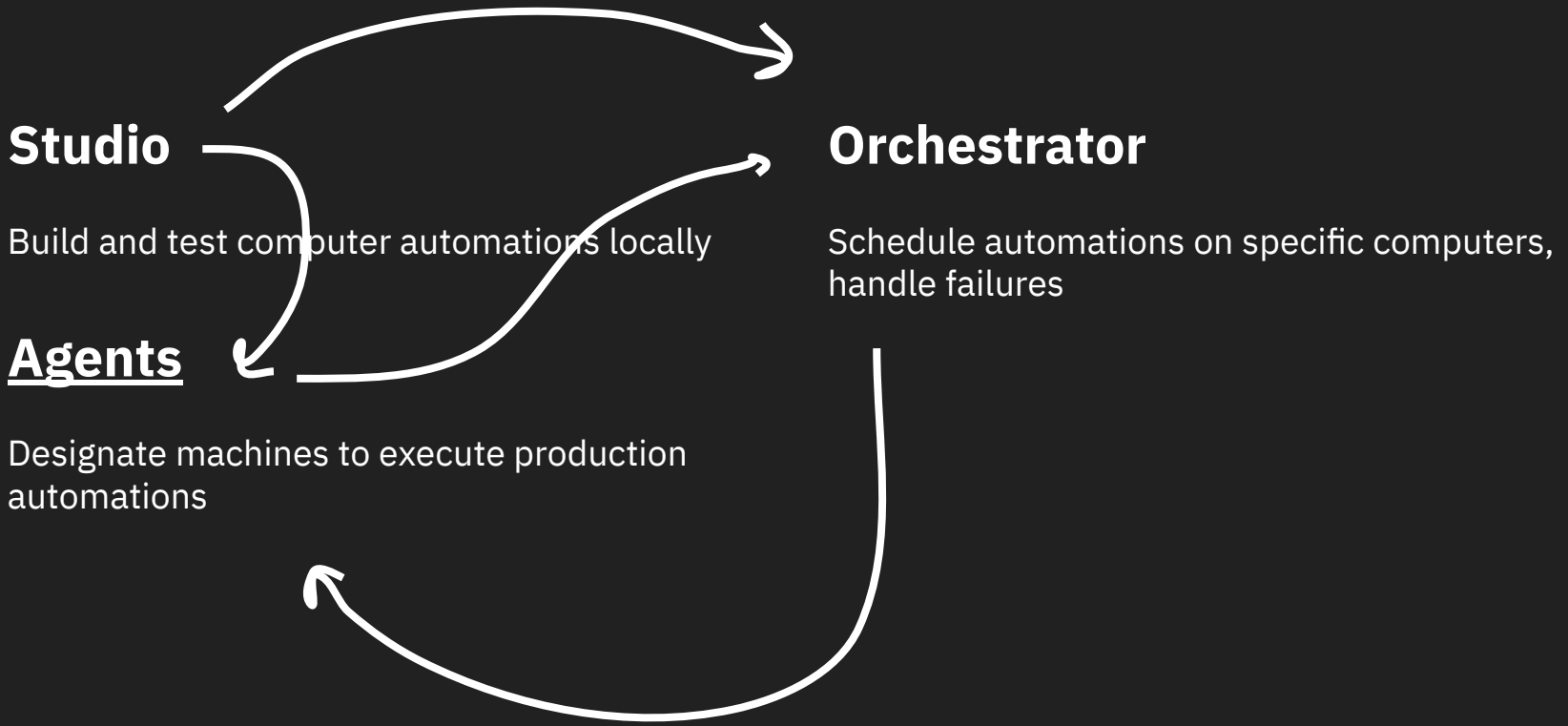
Agents

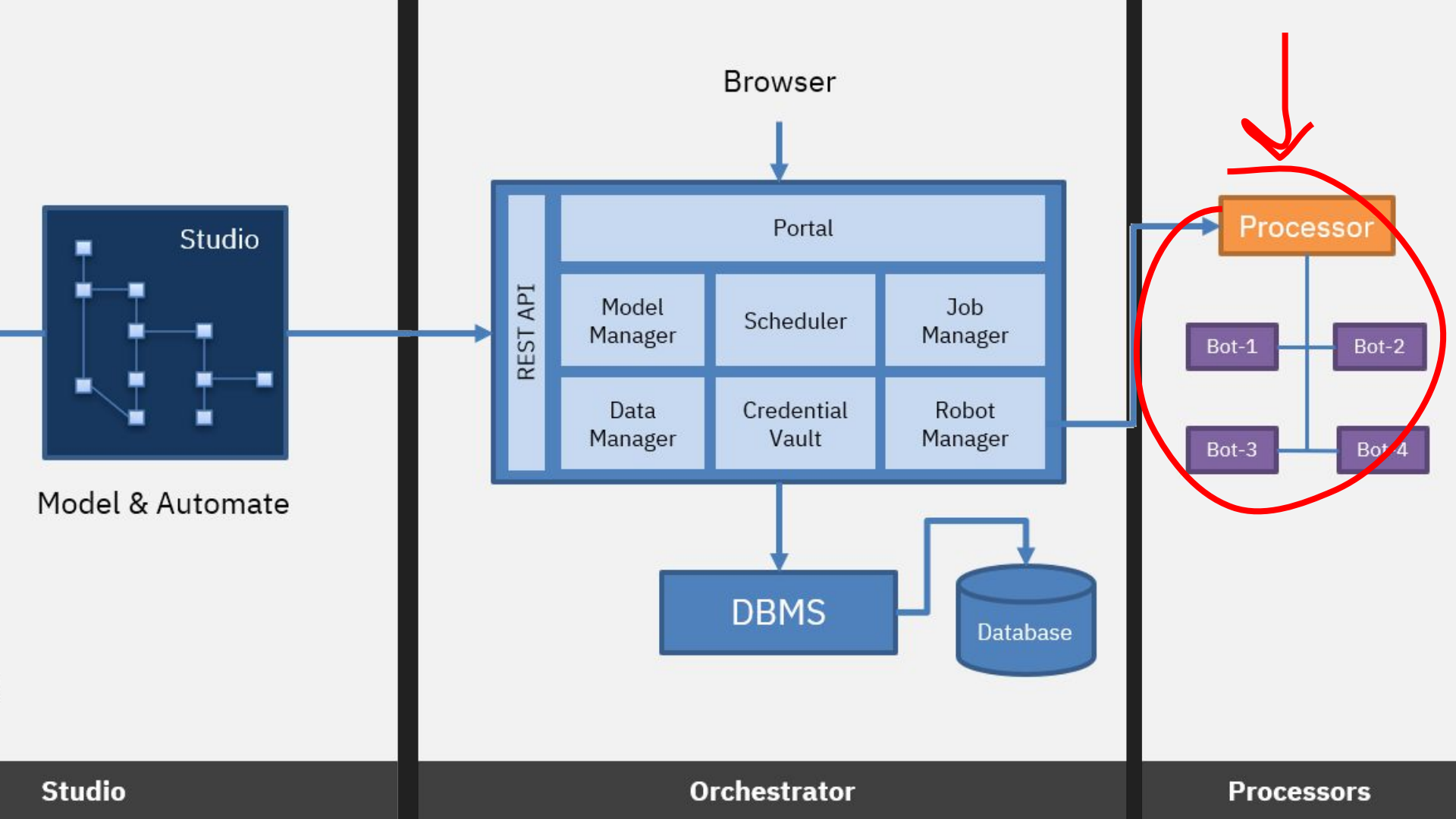
Designate machines to execute production automations

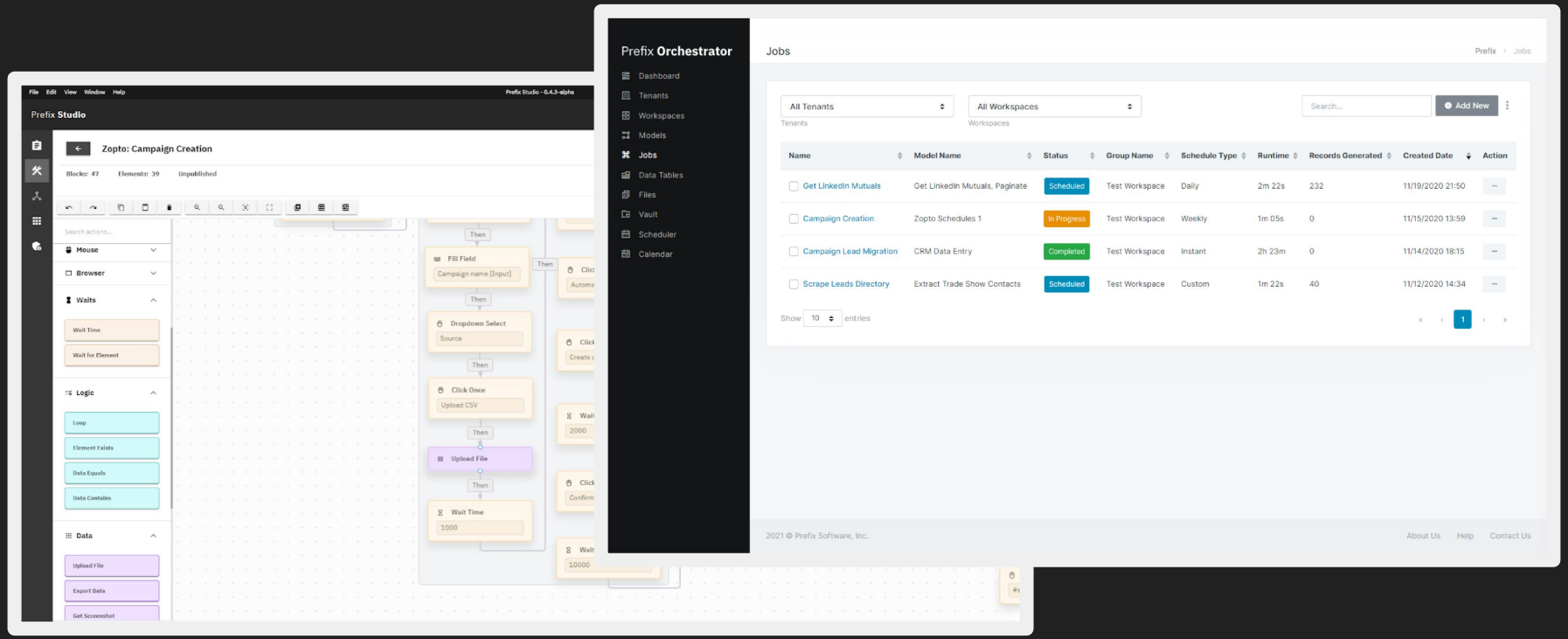
Orchestrator

Schedule automations on specific computers, handle failures

Architecture

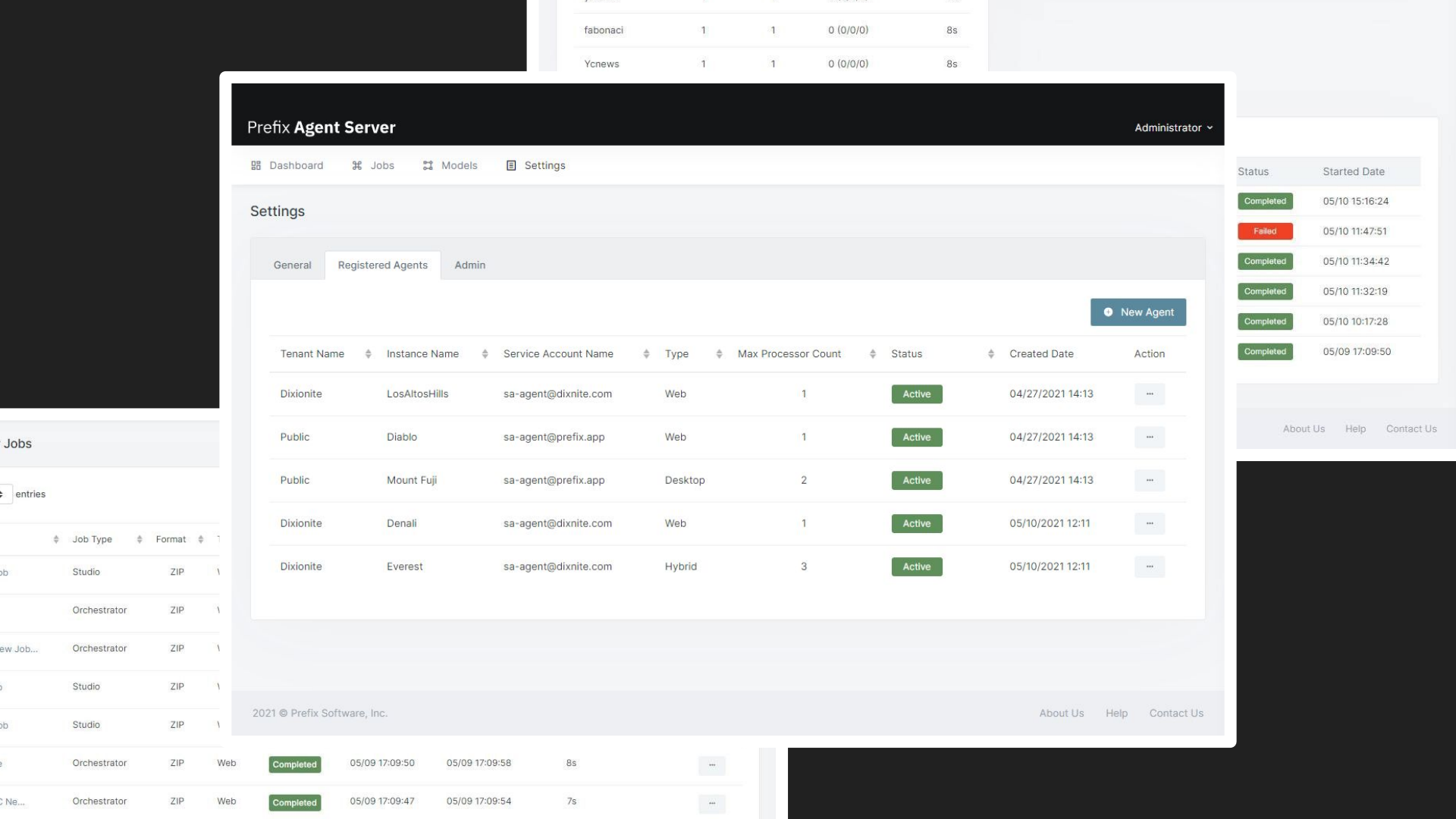



Studio
Model & Automate
Browser
REST API
Portal
Model Manager
Scheduler
Job Manager
Data Manager
Credential Vault
Robot Manager
DBMS
Database
Processor
Bot-1
Bot-2
Bot-3
Bot-4
Studio
Orchestrator
Processors


File Edit View Window Help
Prefix Studio - 0.4.3-alpha
Prefix Studio
Zopto: Campaign Creation
Blocks: 47
Elements: 39
Unpublished
Search actions...
Mouse
Browser
Waits
Wait Time
Wait for Element
Logic
Loop
Element Exists
Data Equals
Data Contains
Data
Upload File
Export Data
Get Screenshot
Then
Fill Field
Campaign name [Input]
Dropdown Select
Source
Click Once
Upload CSV
Upload File
Wait Time
1000
2000
10000
Prefix Orchestrator
Dashboard
Tenants
Workspaces
Models
Jobs
Data Tables
Files
Vault
Scheduler
Calendar
Jobs
Prefix › Jobs
All Tenants
Tenants
All Workspaces
Workspaces
Search...
Add New
Name
Model Name
Status
Group Name
Schedule Type
Runtime
Records Generated
Created Date
Action
Get LinkedIn Mutuals
Get LinkedIn Mutuals, Paginate
Scheduled
Test Workspace
Daily
2m 22s
232
11/19/2020 21:50
Campaign Creation
Zopto Schedules 1
In Progress
Test Workspace
Weekly
1m 05s
0
11/15/2020 13:59
Campaign Lead Migration
CRM Data Entry
Completed
Test Workspace
Instant
2h 23m
0
11/14/2020 18:15
Scrape Leads Directory
Extract Trade Show Contacts
Scheduled
Test Workspace
Custom
1m 22s
40
11/12/2020 14:34
Show 10 entries
1
2021 © Prefix Software, Inc.
About Us
Help
Contact Us

Requirements

1. Parse and execute automation instructions from the Studio
2. Allow users to override currently running automations
3. Connect to the orchestrator to receive automations whenever necessary
4. *Local resource usage metrics*
5. *Failure protocols*


fabonaci 1 1 0 (0/0/0) 8s
Ycnews 1 1 0 (0/0/0) 8s
Prefix Agent Server
Administrator
Dashboard
Jobs
Models
Settings
Settings
General
Registered Agents
Admin
New Agent
Tenant Name
Instance Name
Service Account Name
Type
Max Processor Count
Status
Created Date
Action
Dixionite LosAltosHills sa-agent@dixnite.com Web 1 Active 04/27/2021 14:13
Public Diablo sa-agent@prefix.app Web 1 Active 04/27/2021 14:13
Public Mount Fuji sa-agent@prefix.app Desktop 2 Active 04/27/2021 14:13
Dixionite Denali sa-agent@dixnite.com Web 1 Active 05/10/2021 12:11
Dixionite Everest sa-agent@dixnite.com Hybrid 3 Active 05/10/2021 12:11
2021 © Prefix Software, Inc.
About Us
Help
Contact Us
Status
Started Date
Completed 05/10 15:16:24
Failed 05/10 11:47:51
Completed 05/10 11:34:42
Completed 05/10 11:32:19
Completed 05/10 10:17:28
Completed 05/09 17:09:50
About Us
Help
Contact Us
Jobs
entries
Job Type
Format
Studio ZIP
Orchestrator ZIP
Orchestrator ZIP
Studio ZIP
Studio ZIP
Orchestrator ZIP Web Completed 05/09 17:09:50 05/09 17:09:58 8s
Orchestrator ZIP Web Completed 05/09 17:09:47 05/09 17:09:54 7s

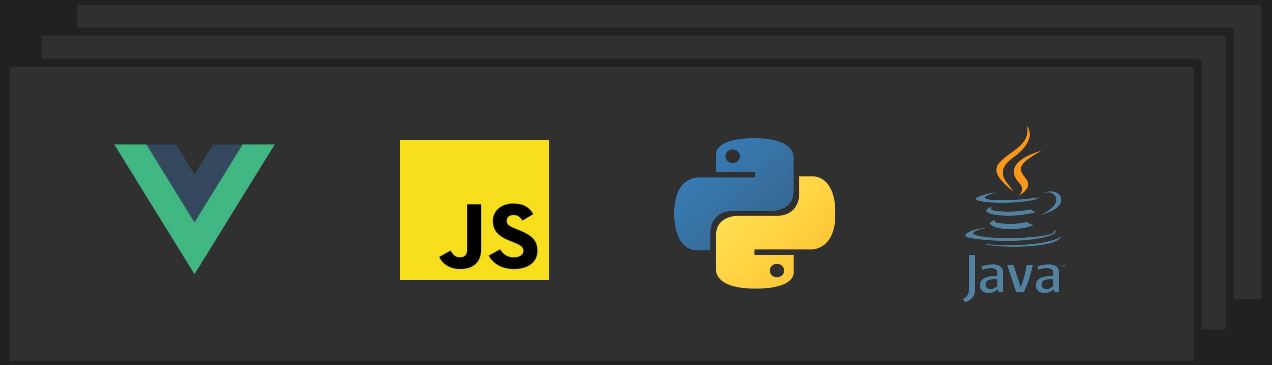
Demo

Mechanism

1. An account is provisioned on the Orchestrator for the Agent Server
2. The Agent Server logs into the Orchestrator via this service account and pings the Orchestrator every X seconds for any new automations
3. If a new automation task is available, the Agent Server makes a request to the Orchestrator for the automation instructions
4. The Agent Server downloads automation info, executes the instructions, and returns its results to the Orchestrator

Twists

1. Agent Servers configured incorrectly DDoS’d the orchestrator
2. Limit concurrent automations
3. Agent Servers may be deployed behind a firewall
4. “E-Stop” on automations for a particular machine


JS
Java

Improvements

1. Determine when Agent Servers are being run headlessly
    a. Filter automations executed by compatibility
2. Fix/document confusing nomenclature
3. Native studio integration with Agents
    a. Studio (development) -> Agent (staging) -> Orchestrator (production)
4. Wrap application in an installer
5. Develop load tests

End