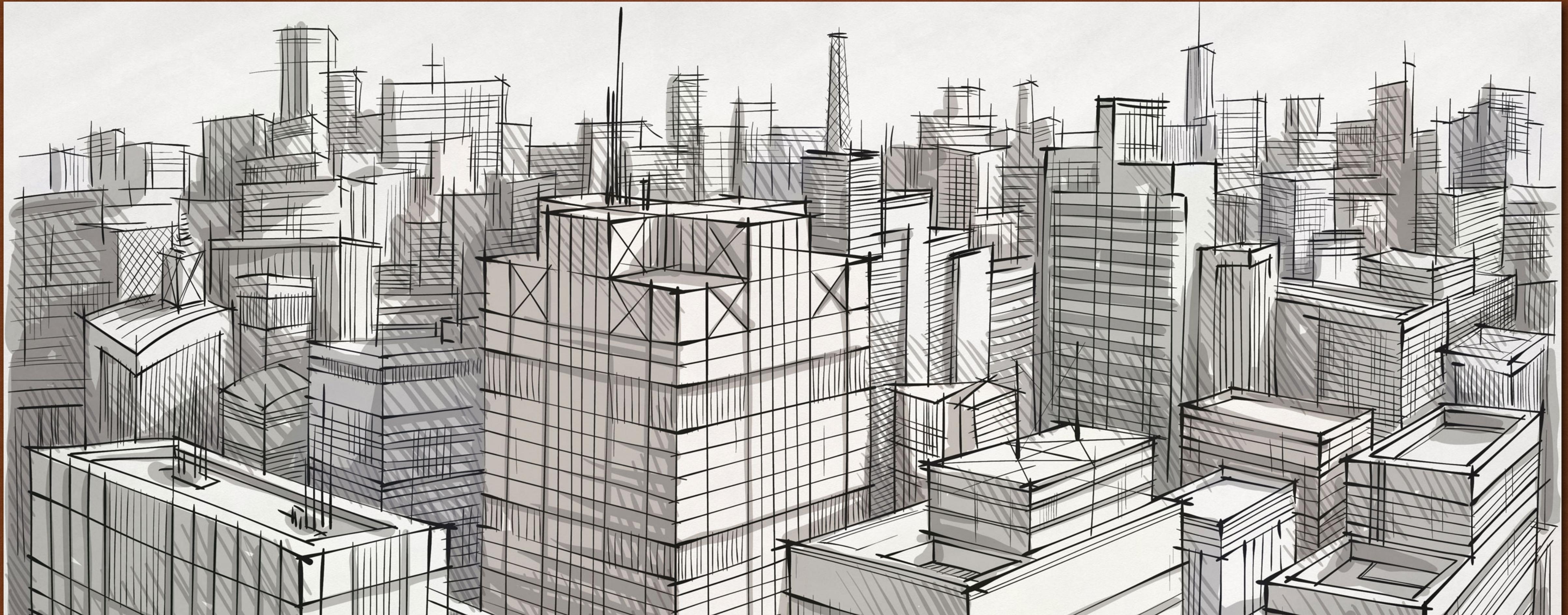


StageZero Launch

Build it fast, without compromising quality



The house built on sand cannot stand

StageZero

Your product is only as good as your engineering team

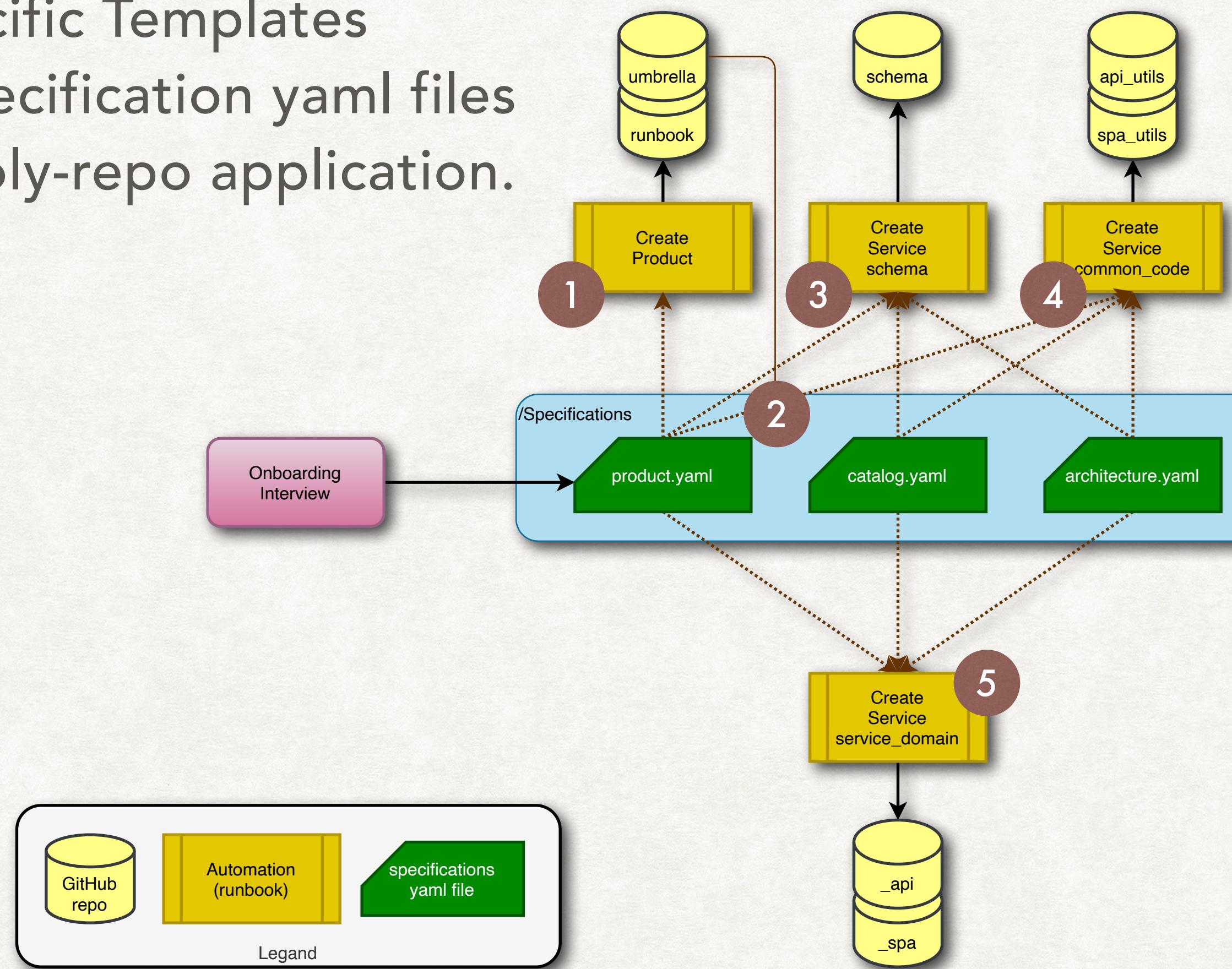
- Developer Experience is a key ingredient for a profitable product.
- Software craftsmanship frames a delightful developer experience.
- Vibecoding does not mean no software engineers, and it should not compromise on craftsmanship.
- Product launch, or the time from idea to MVP release, is the biggest graveyard in software.



StageZero

Engineering Launch Journey

Technology Specific Templates
Merged with Specification yaml files
to generate a poly-repo application.



StageZero

Umbrella Repo

README.md - Product overview

CONTRIBUTING.md - Developer Onboarding process

/DeveloperEdition

SRE maintained CLI (zsh)wraps ``docker compose`` commands

Standards documents for Data, API, UI, and SRE engineering

/Research

A place to share research findings

/Specifications

yaml files are harvested from design workshops

Technology Agnostic product specifications (product.yaml, catalog.yaml)

Technology Specific architecture.yaml document

/Tasks

A folder for common coding tasks

Structured to support long-running LLM jobs

/Workshops

Raw design yaml files collected during design workshops

Pre-cursor to Specification yaml files

StageZero

Runbooks Repo

Runbooks are just Markdown files,
Code blocks for setup and script
Container packaging of automations
Runbook execution in isolated sandbox
Server Records executions outputs*
Script dependencies pre-installed:

- zsh, jq, yq, curl, git
- gh (with provided credentials)
- ssh (with provided credentials)
- docker (with provided credentials)
- other dependencies can be added.

Can be secured to deploy on remote infrastructure.

*logged, and added to history in runbook. Data is ephemeral unless persisted outside the container.

```
# Sample Runbook

This is a simple test runbook that demonstrates the basic functionality of the stage0_runner utility.
It performs a simple echo operation and lists the current directory.

# Environment Requirements
```yaml
TEST_VAR: A test environment variable for demonstration purposes
```

# File System Requirements
```yaml
Input:
```

# Required Claims
```yaml
roles: sre, api, ux, data
```

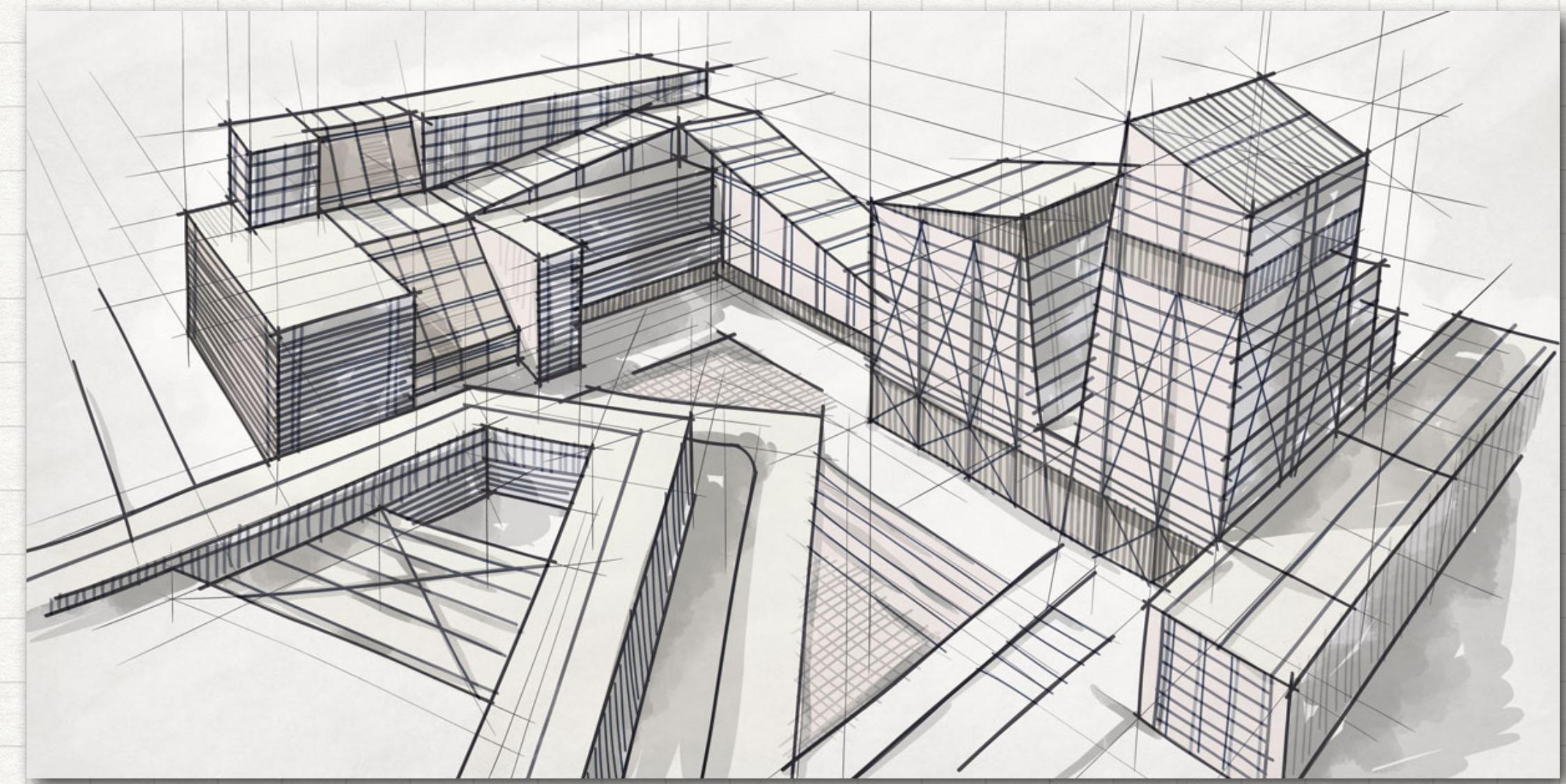
This section is optional. If present, the token must include the specified claims to execute or validate the runbook.
- `roles`: List of roles (comma-separated) that are allowed to execute/validate this runbook
- Other claims can be specified as key-value pairs where the value is a comma-separated list of allowed values

# Script
```sh
#!/bin/zsh
echo "Running SimpleRunbook"
echo "Test variable value: ${TEST_VAR:-not set}"
echo "Current directory: $(pwd)"
echo "Listing files:"
ls -la
echo "SimpleRunbook completed successfully"
```

# History
```

StageZero Post-Launch

- Launch is not orbit
- Generated code is:
 - Functional - it does “something”
 - Well Crafted - well tested
 - Packaged for Automated CI
 - A starting point for rapid iteration work
- Generated code is NOT:
 - highly customized or branded.
 - a MVP ready for production deployment.
 - everything that you need.



- Future Stage0 Services - Post Launch “Mission Control”
 - Incorporate Long-Running LLM Workflow in Automations
 - Add templates with async pub/sub data domains
 - Add templates / runbooks for cloud deployment (Terraform?)
 - Expose Runbooks as MCP Server for LLM backed Discord Bot
 - Build Discord Bot Workflows for Design
(Client/Product Onboarding, Stage0 Launch, Design Workshops etc.)
 - Build Discord Bot Workflows for Product Management
(Release automation, test scheduling etc.)
 - Build Discord Bot Workflows for Product Support
(service management, incident response, etc.)

StageZero Demo Prep

Think of a new Product Name and Description.

Create a GitHub Organization for us to play in at <https://github.com/organizations/plan> - The Free Plan is fine, please use an all_lower_case name. Invite FlatBallFlyer to the organization with Owner privelages.

Create and save a GITHUB_TOKEN so you can run automations to create and update source code repo's and packages.

GitHub.com -> Profile Pic -> Settings -> Developer Settings -> Personal access tokens -> Tokens(classic) -> Create New -> repo, workflow, write:packages privelages.

Install pre-requisites:

- **WSL** - For windows users - <https://learn.microsoft.com/en-us/windows/wsl/install>
- **Docker Desktop** - <https://www.docker.com/get-started/>
- **zsh shell** - Default on MacOS, <https://github.com/ohmyzsh/ohmyzsh/wiki/Installing-ZSH> otherwise
- **jq** - Yaml Parser, ``brew install jq`` on Mac or <https://mikefarah.gitbook.io/jq>

StageZero Demo

Step1: Create Product Umbrella and Runbook

Step2: Update your product specifications

Step3: Install the developer CLI

Step4: Generate all the code

Step5: Run your functional app

Get access to Mikes Runbook repo to bootstrap your project, then

```
touch ~/playground/delete_me
cd ~/playground
git clone git@github.com:FlatBallFlyer/Runbooks.git
cd Runbooks/runbooks
./stage0_demo.sh
```

The script will prompt for a few values, and then guide you through the demo. The script doesn't "do" anything, but it will provide copy/paste-able commands when needed.