

Vitest and the Anatomy of a Test Runner

Hello! My name is Jessica Sachs.
I go by jess. she/her pronouns.

- Work in the Vue, Vite, and Vitest testing ecosystem
- Staff Engineer at Cypress where I lead the development of the Cy Component Test Runner
- Faker.js maintainer



What's  **Vitest ?**

**Vitest is a new testing
framework built by many of the
Vue Core team members to
support testing Vite-based apps**

Vitest is a new testing framework built by many of the Vue Core team members to support testing Vite-based apps



Vitest is a new testing framework built by many of the Vue Core team members to support testing Vite-based apps

Recommendation

- Vitest

Since the official setup created by `create-vue` is based on **Vite**, we recommend using a unit testing framework that can leverage the same configuration and transform pipeline directly from Vite. **Vitest** is a unit testing framework designed specifically for this purpose, created and maintained by Vue / Vite team members. It integrates with Vite-based projects with minimal effort, and is blazing fast.





/bin/zsh

Get started with Vitest + Vue today!

```
npm init vue
```



**Vite replaces Vue CLI
(webpack) as the new
recommended bundler for Vue
Apps**



**Vite works with Vue 2 via
underfin/vite-plugin-vue2, and
so does Vitest**

vite-plugin-vue2 

1.9.3 • **Public** • Published 21 days ago

Vitest is framework agnostic...

...but because it's built by **Vue,
Nuxt, and **Vite Core** team
members, it's architected with
**Vue (and Svelte!) support in
mind.****

Okay, but what's  Vitest really?

Okay, but what's a test runner really?

Okay, but what's a test runner really?

A program



/bin/zsh

Start the Vitest Test Runner

npm run vitest

Command Line API



test-runner.js

```
1 import { startVitest } from 'vitest'
2 const result = await startVitest(/* options */)
3 // *This API doesn't exist, for demo only
```

Node API

Okay, but what's a test runner really?

A **program** that executes **tests** against your source code



```
1 it('renders the message passed in', () => {
2   const wrapper = mount(HelloWorld, {
3     props: {
4       message: 'Hello from Boston!',
5     },
6   })
7
8   expect(wrapper.text()).toContain('Hello from Boston!')
9 })
```

Test Code

Okay, but what's a test runner really?

A **program** that executes **tests** against your source code such that if your code doesn't do what you want it to, it **reports any failures**.



/bin/zsh

```
→ todo-app pnpm vitest run --environment jsdom --reporter verbose  
RUN /Users/jess/projects/todo-app
```

```
✓ src/components/__tests__/HelloWorld.spec.js (1)  
  ✓ HelloWorld (1)  
    ✓ renders properly
```

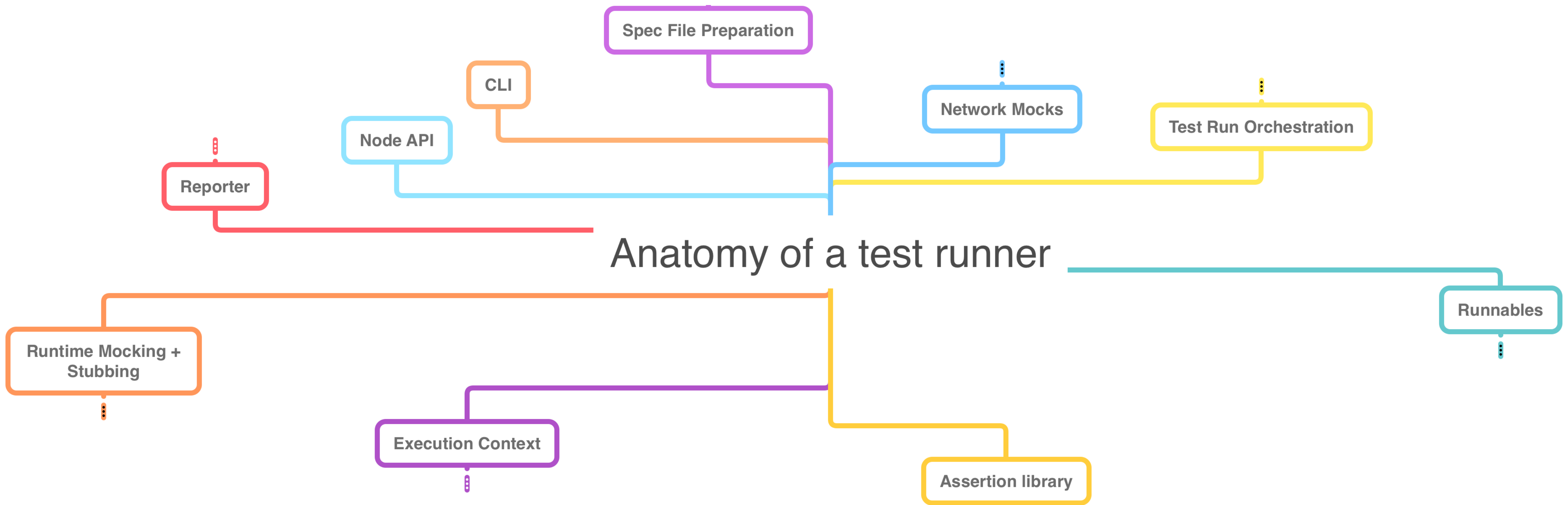
```
Test Files  1 passed (1)  
Tests      1 passed (1)  
Time       934ms (in thread 9ms, 10660.26%)
```

Reporter

@_JessicaSachs

The simplest test runner

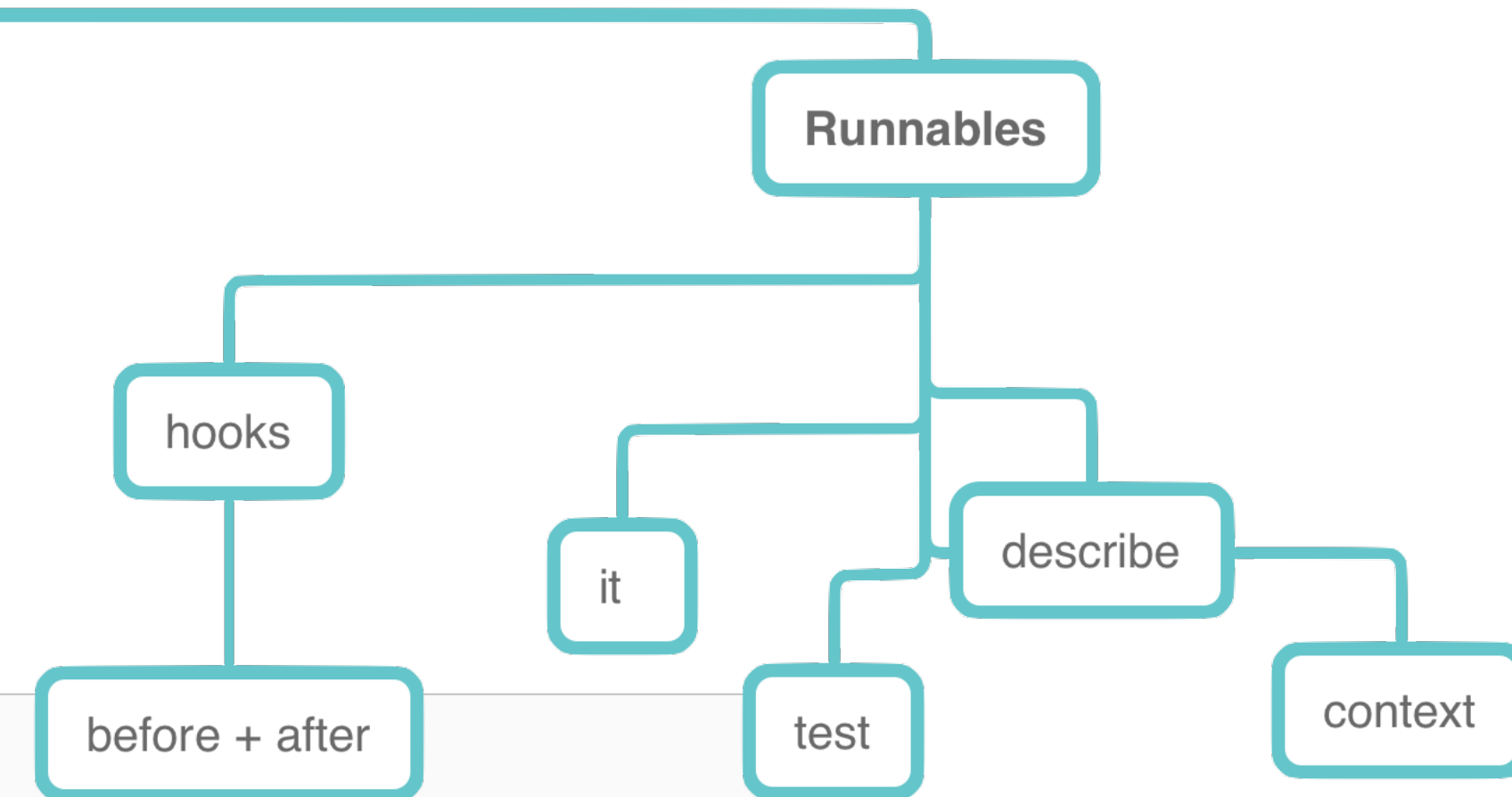
- Builds your spec's code and any source code imported
- Invokes specs to create a **Test Plan**
- Executes the Test Plan
- Collects passes, failures, and any “skipped” tests
- Process exits with a success or error status code



Anatomy of a test runner

Execution Context

Assertion library

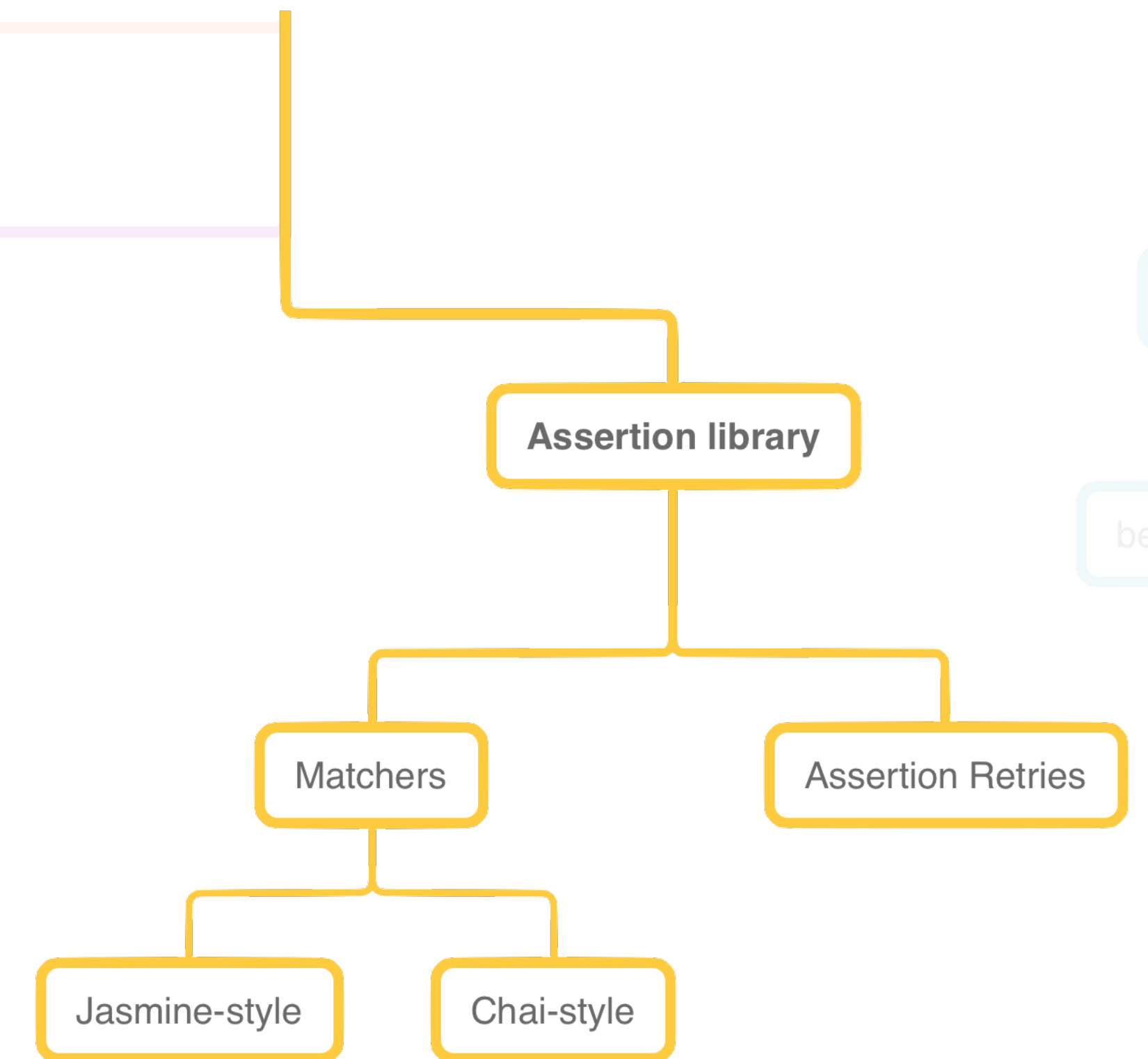


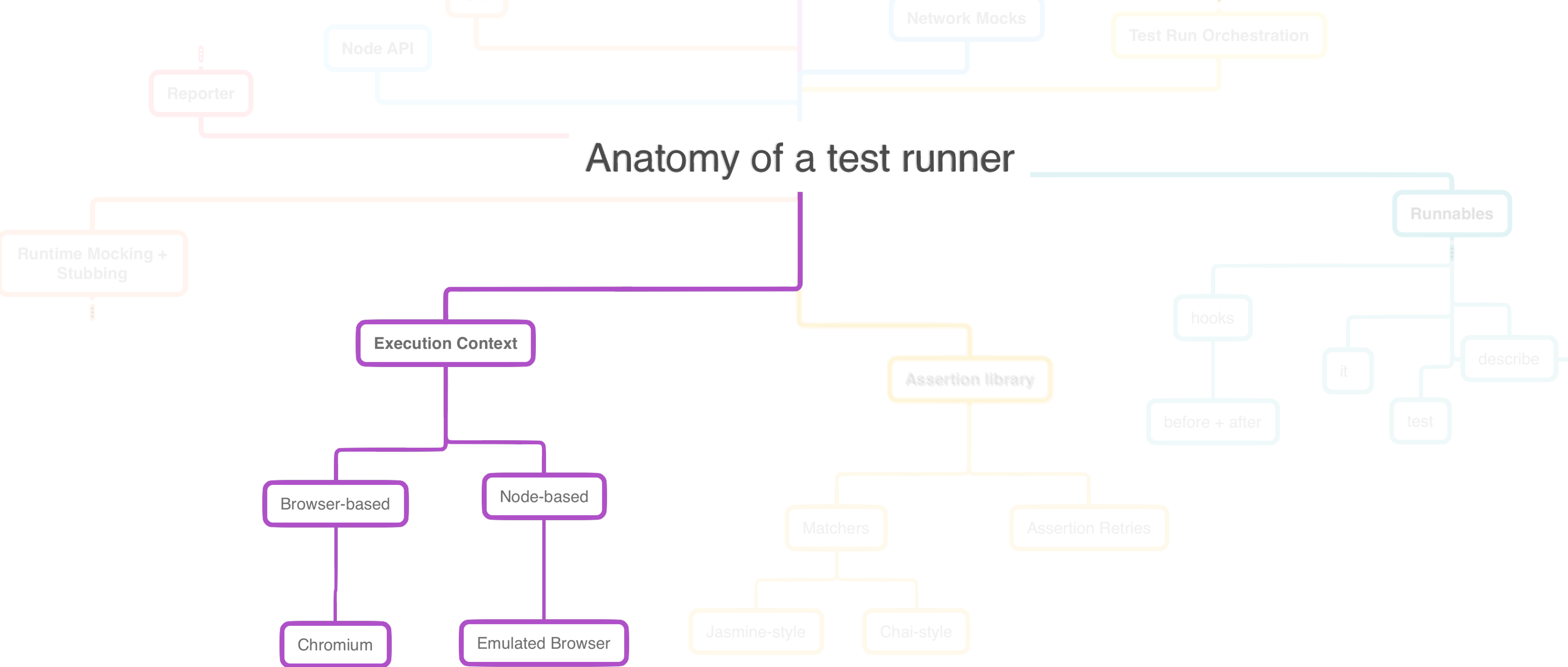
```
1 interface Runnable {
2   description?: string
3   body: Function
4 }
5
6 interface Hook extends Runnable {
7   type: 'before' | 'after' | 'beforeEach' | 'afterEach'
8 }
9 interface Test extends Runnable {} // it or test
10 interface Suite extends Runnable {} // describe or context
```

Anatomy of a test runner

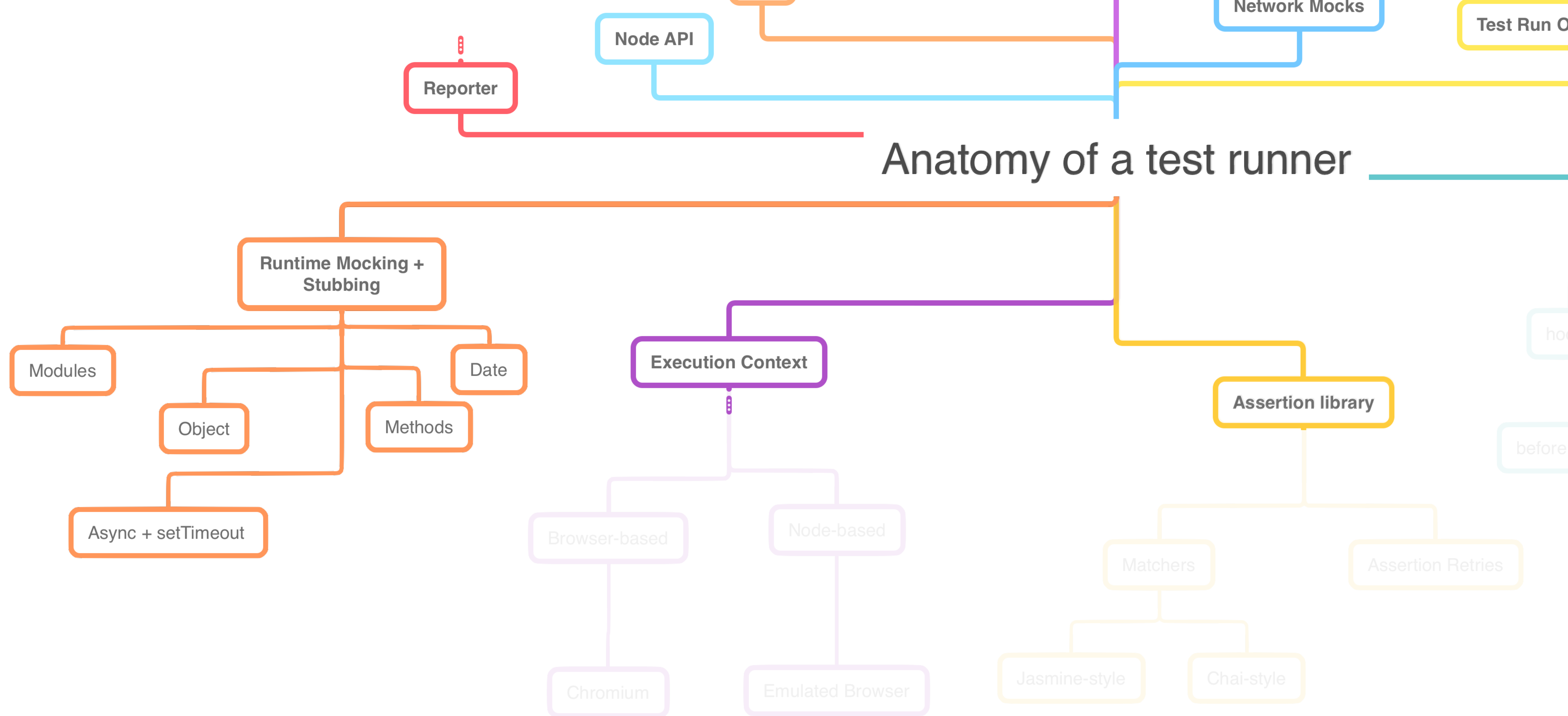
An assertion library, like Chai or Jasmine, is responsible for implementing matchers like the below:

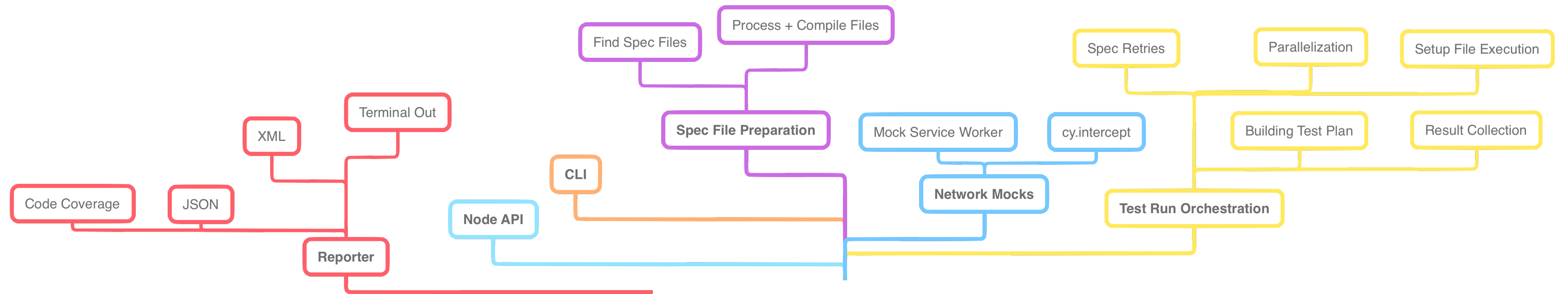
- toContain
- toMatch
- toBeVisible
- toBeTruthy
- toBeCalledWith
- toBeLessThan
- to.contain
- to.match
- to.be.visible
- to.be.true
- to.be.called.with
- to.be.less.than



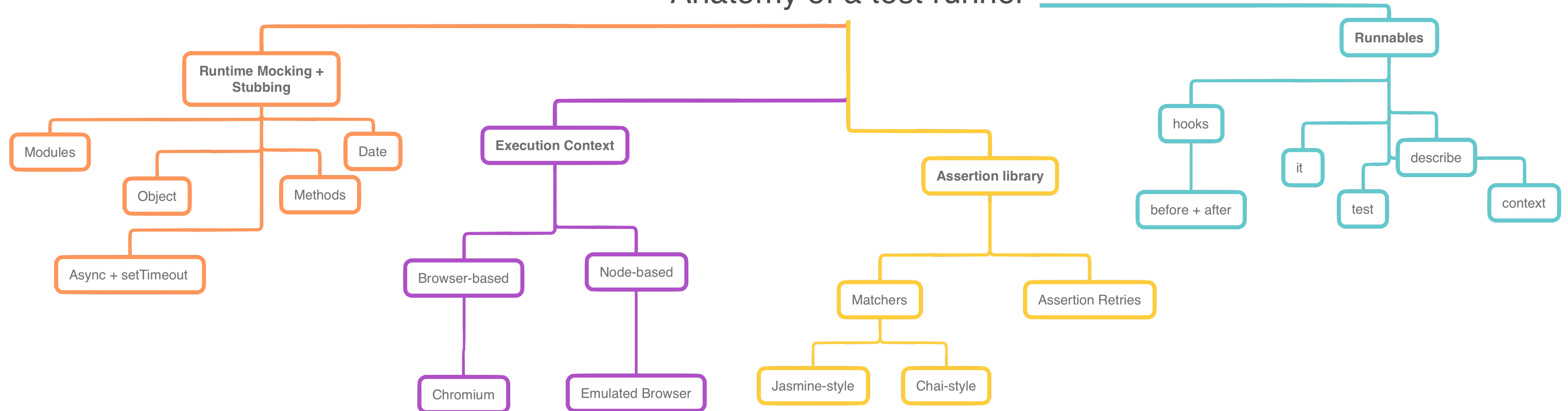


Anatomy of a test runner





Anatomy of a test runner



Why Vitest?

**It's fast. Very, very fast.
Watch mode leverages Vite's HMR.**

Unlike other JavaScript test runners, it runs on top of Go.



Vitest



Vite



esbuild
An extremely fast JavaScript bundler



Why Vitest?

It's maintainable. It re-uses your Vite build config, so it works out of the box with your Vue app.

Why Vitest?

It keeps you safe.

Testing a Vite app using Mocha or Jest requires you to reimplement your build in Webpack or Jest transformers.

This means when you're testing your Vite app with a non-Vite build, you're not testing the source code you're going to ship.

Whenever test !== prod you risk releasing bugs that are *covered up* by your “test-only” config.



jess

@_jessicasachs



Replying to [@AdamRackis](#) [@berndschimmer](#) and [@css](#)

The big thing that [@vite_dev](#) gives you is consistency in execution environments between test and prod. Maintaining two sources of truth for the asset pipelines, module resolution, and env leads to bugs not being caught by tests.

9:50 PM · Jan 11, 2022 · Twitter for iPad

2 Retweets **1** Quote Tweet **34** Likes

@_JessicaSachs

Additionally...



jess

@_jessicasachs



Replying to @_jessicasachs @AdamRackis and 3 others

Vite-specific features like Virtual File Resolution are essentially incompatible with Jest unless you take on a lot of work. vitejs.dev/guide/api-plugin...

9:59 PM · Jan 11, 2022 · Twitter for iPad

And again...

It's built by the **Vue**, **Nuxt**, and **Vite Core** team members, so it's architected with Vue support in mind.

So, why  *Vitest* ?

It's fast.

It's maintainable.

It keeps you safe.

It's Vue core-built and maintained.



Thank you!

jess

@_jessicasachs

Building OSS every day. Architect of Cy Component Testing [@cypress_io](#). Vue team member. Maintaining [@faker_js](#).

📍 Cambridge, MA 📅 Joined October 2018

1,943 Following **4,554** Followers

Slides are available at jess.sh/vitest-anatomy

@_JessicaSachs