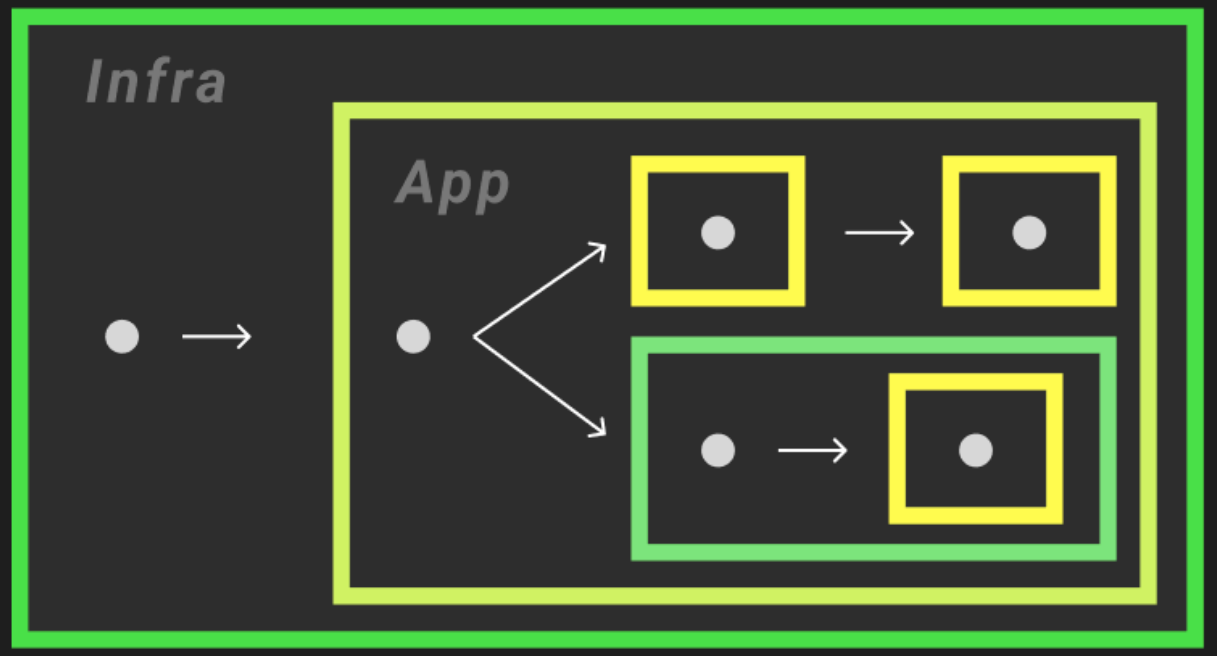
The SOFTWARE ESSENTIALIST

Physical Essential: Composition



Physical Essential

Composition

Software is made from a lot of different parts, but many developers wire these parts up without much thought.

It matters. A lot.

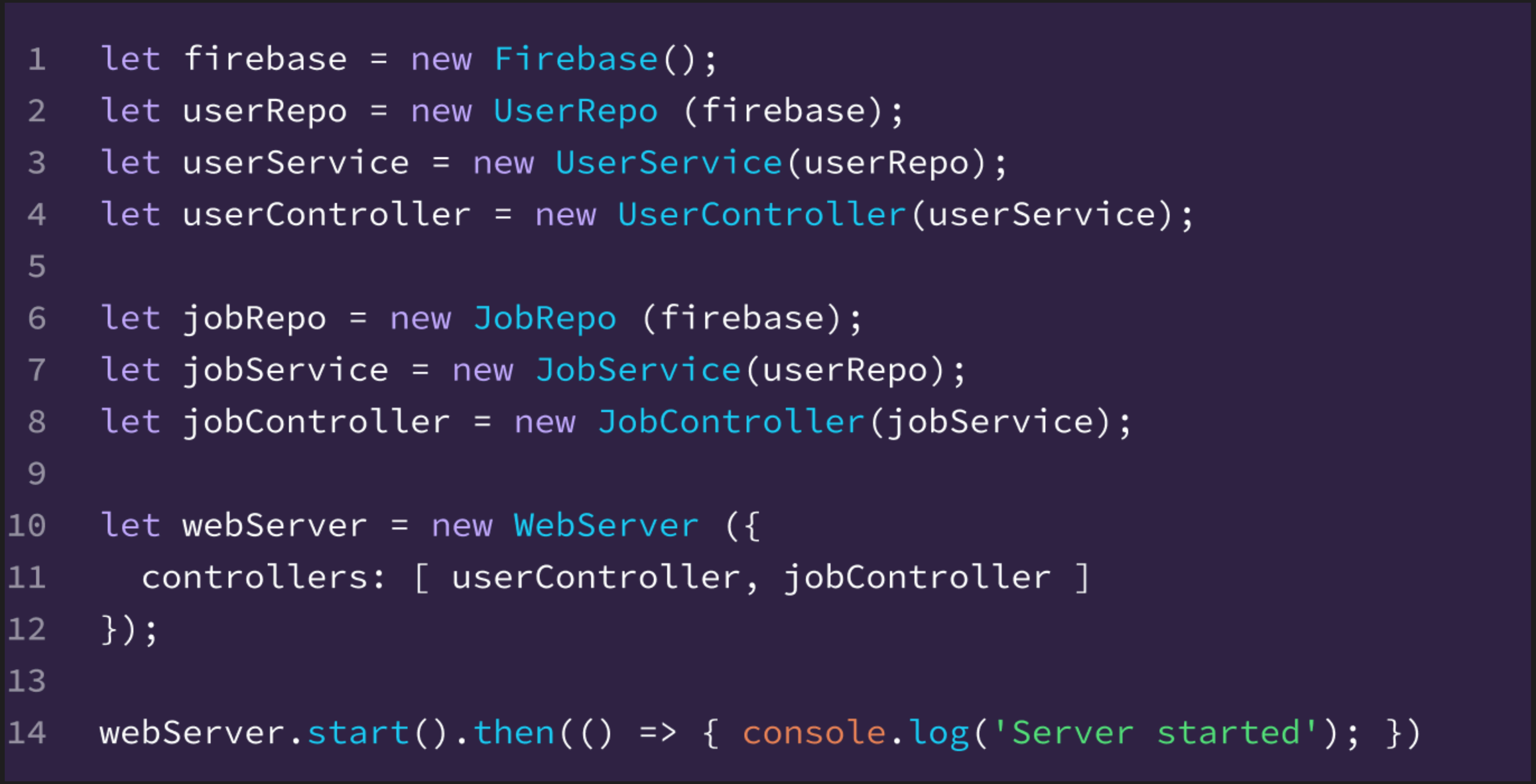
Without conscious thought, you can create a coupled codebase where abstractions are difficult to use, understand, test, and maintain.

Composition is about the way you consciously create a network of collaborating abstractions.

When done well, your application is configurable from a single location and it is easy to set up dynamic runtime behavior for different environments: dev, prod, testing, ci, and so on.

```javascript
if (env === 'dev') {
 // do something
} else if (env === 'prod') {
 // do this instead
}
```

essentialist.dev

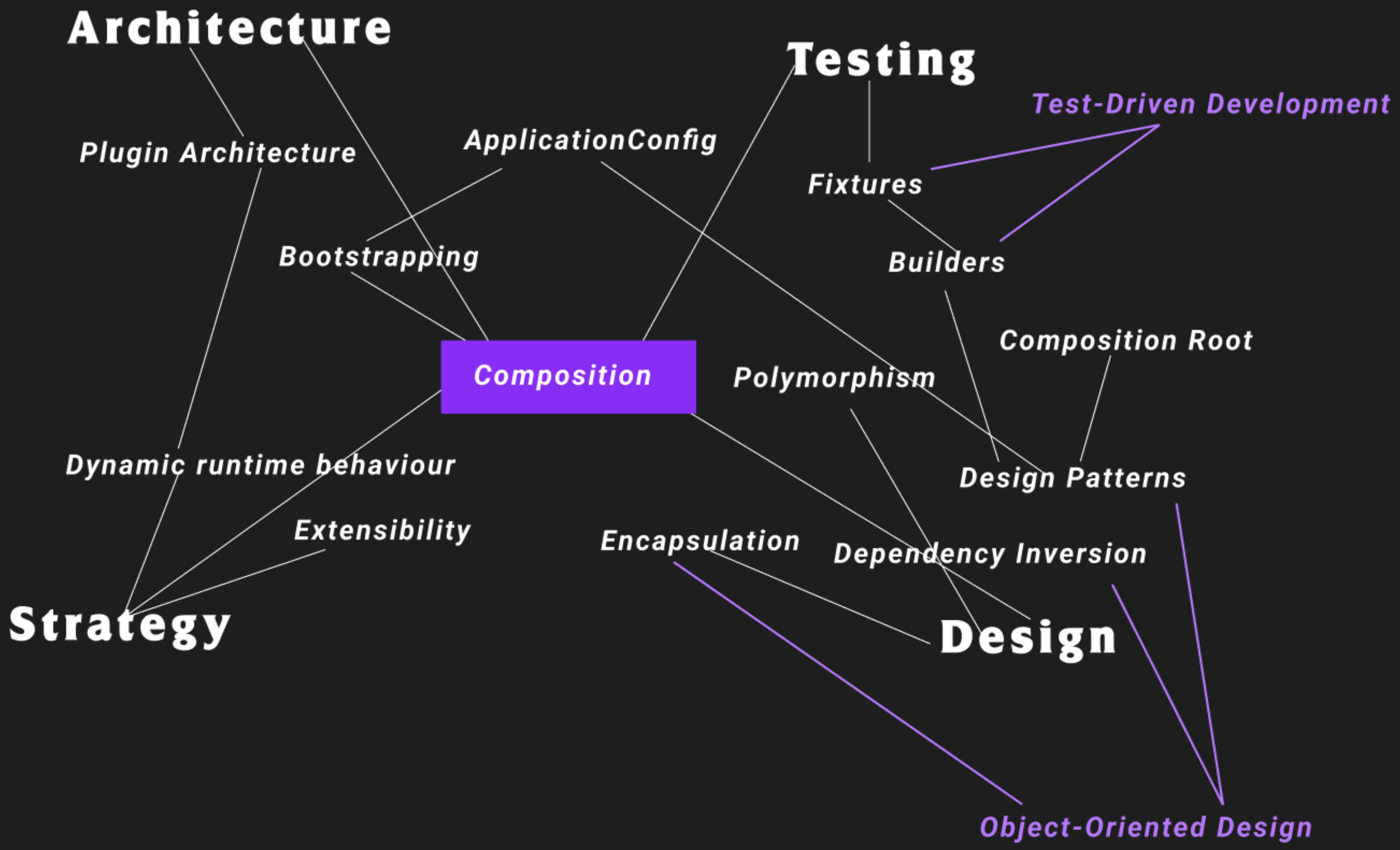


```javascript
let firebase = new Firebase();
let userRepo = new UserRepo (firebase);
let userService = new UserService(userRepo);
let userController = new UserController(userService);

let jobRepo = new JobRepo (firebase);
let jobService = new JobService(userRepo);
let jobController = new JobController(jobService);

let webServer = new WebServer ({
  controllers: [ userController, jobController ]
});

webServer.start().then(() => { console.log('Server started'); })
```

essentialist.dev

```
let root = new CompositionRoot();

root.setContext('dev');

let webServer = root.getWebServer();

webServer.start().then((message) => { console.log(message); })

// Server started in dev mode
```

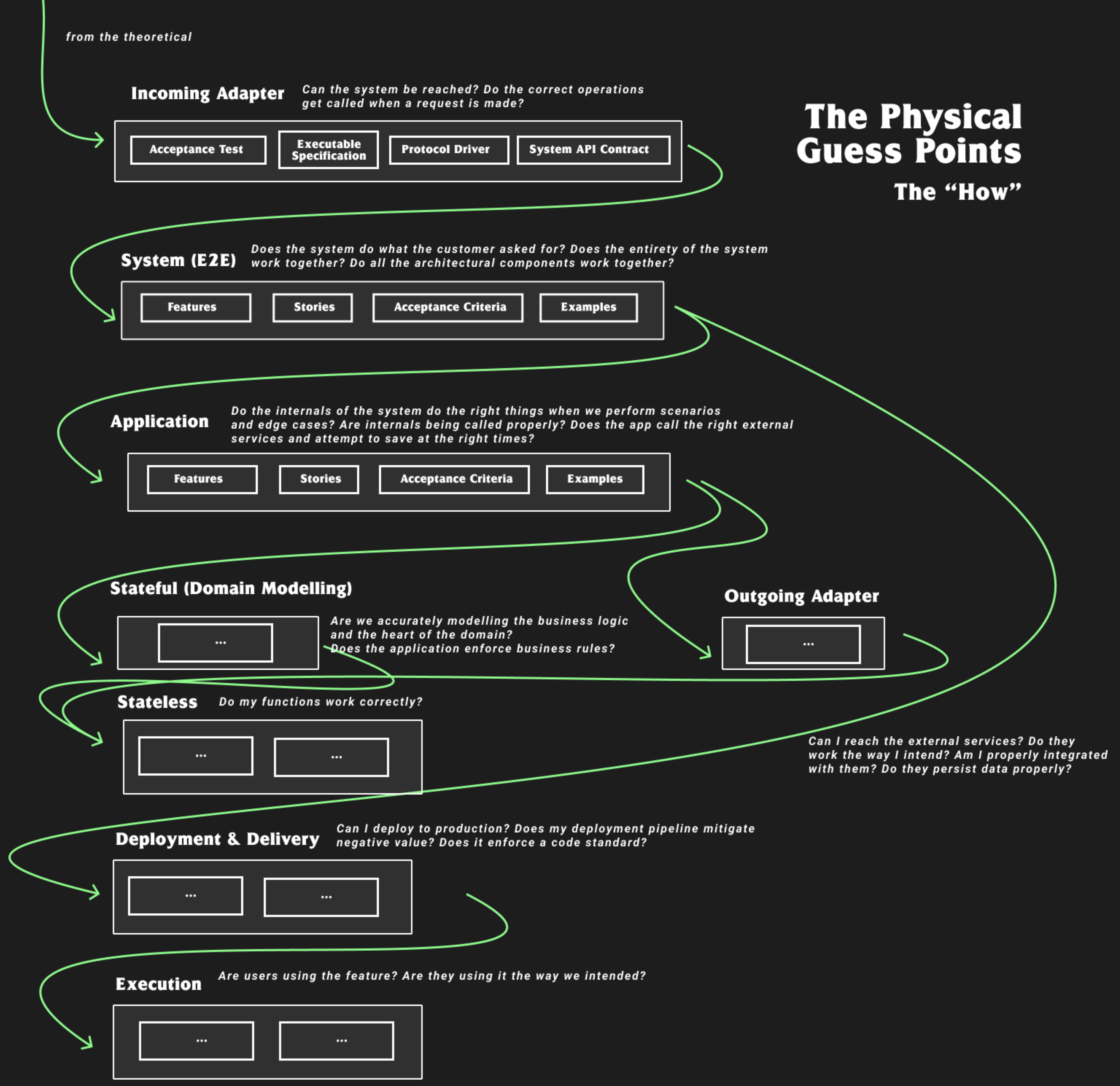
essentialist.dev



essentialist.dev

The Physical Guess Points

The “How”



Incoming Adapter: *Can the system be reached? Do the correct operations get called when a request is made?*

System (E2E): *Does the system do what the customer asked for? Does the entirety of the system work together? Do all the architectural components work together?*

Application: *Do the internals of the system do the right things when we perform scenarios and edge cases? Are internals being called properly? Does the app call the right external services and attempt to save at the right times?*

Stateful (Domain Modelling): *Are we accurately modelling the business logic and the heart of the domain? Does the application enforce business rules?*

Outgoing Adapter: *Can I reach the external services? Do they work the way I intend? Am I properly integrated with them? Do they persist data properly?*

Stateless: *Do my functions work correctly?*

Deployment & Delivery: *Can I deploy to production? Does my deployment pipeline mitigate negative value? Does it enforce a code standard?*

Execution: *Are users using the feature? Are they using it the way we intended?*

essentialist.dev

Where we’ll learn more