



WEB DEVELOPMENT

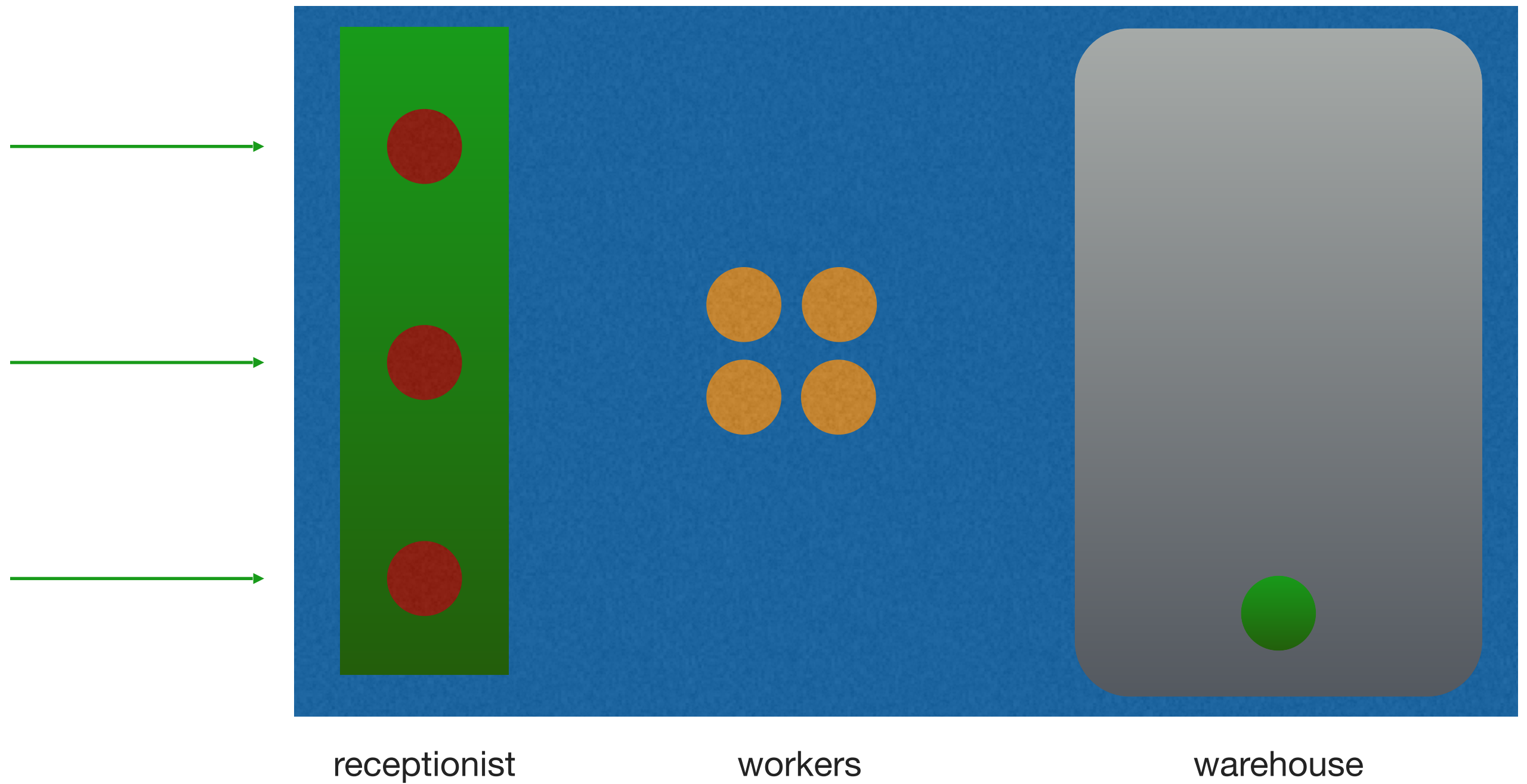
Lesson 0

What is a Web Application?

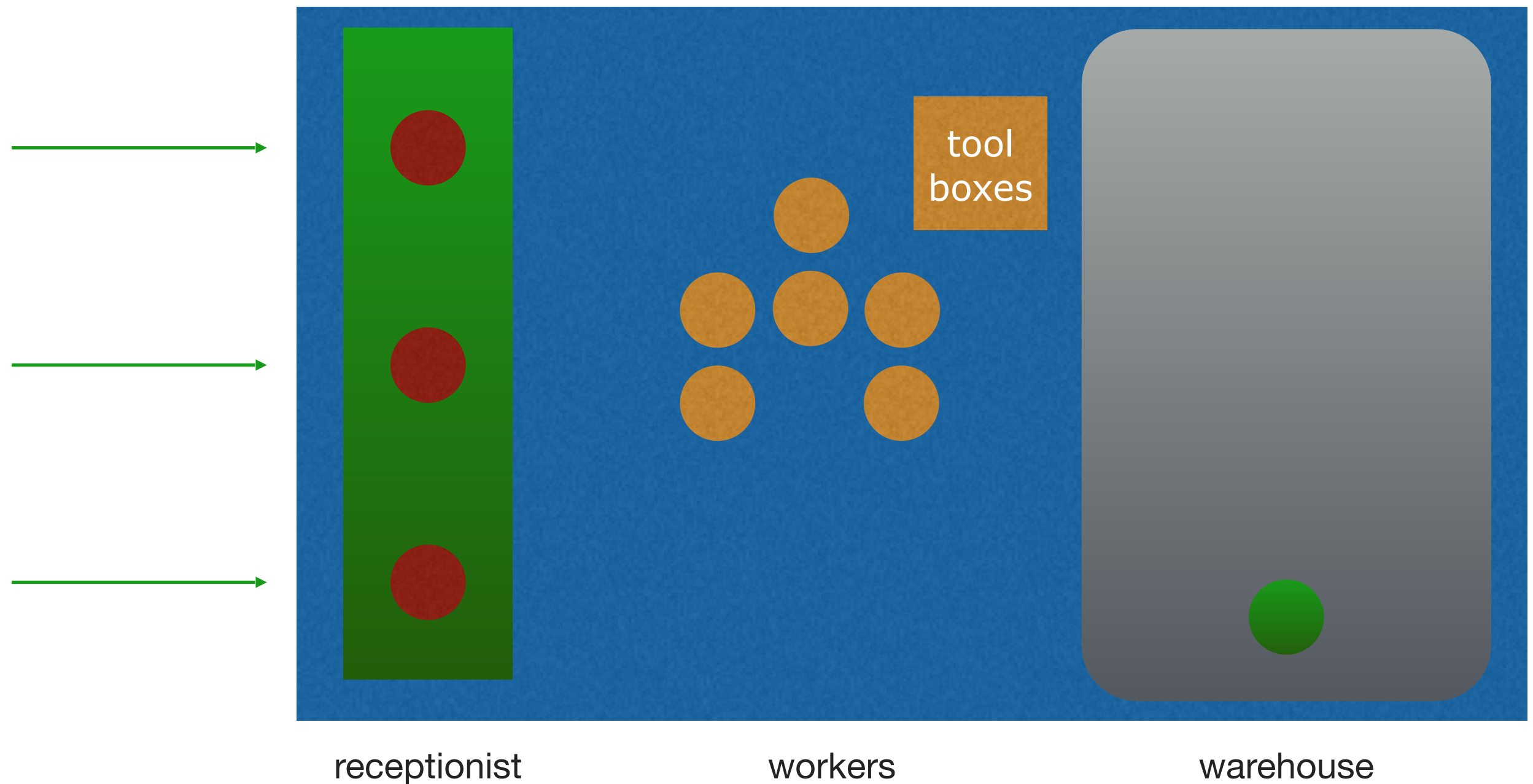
A web application is software that runs on a web server and is accessed via a browser, without needing installation.

How does the Web work?

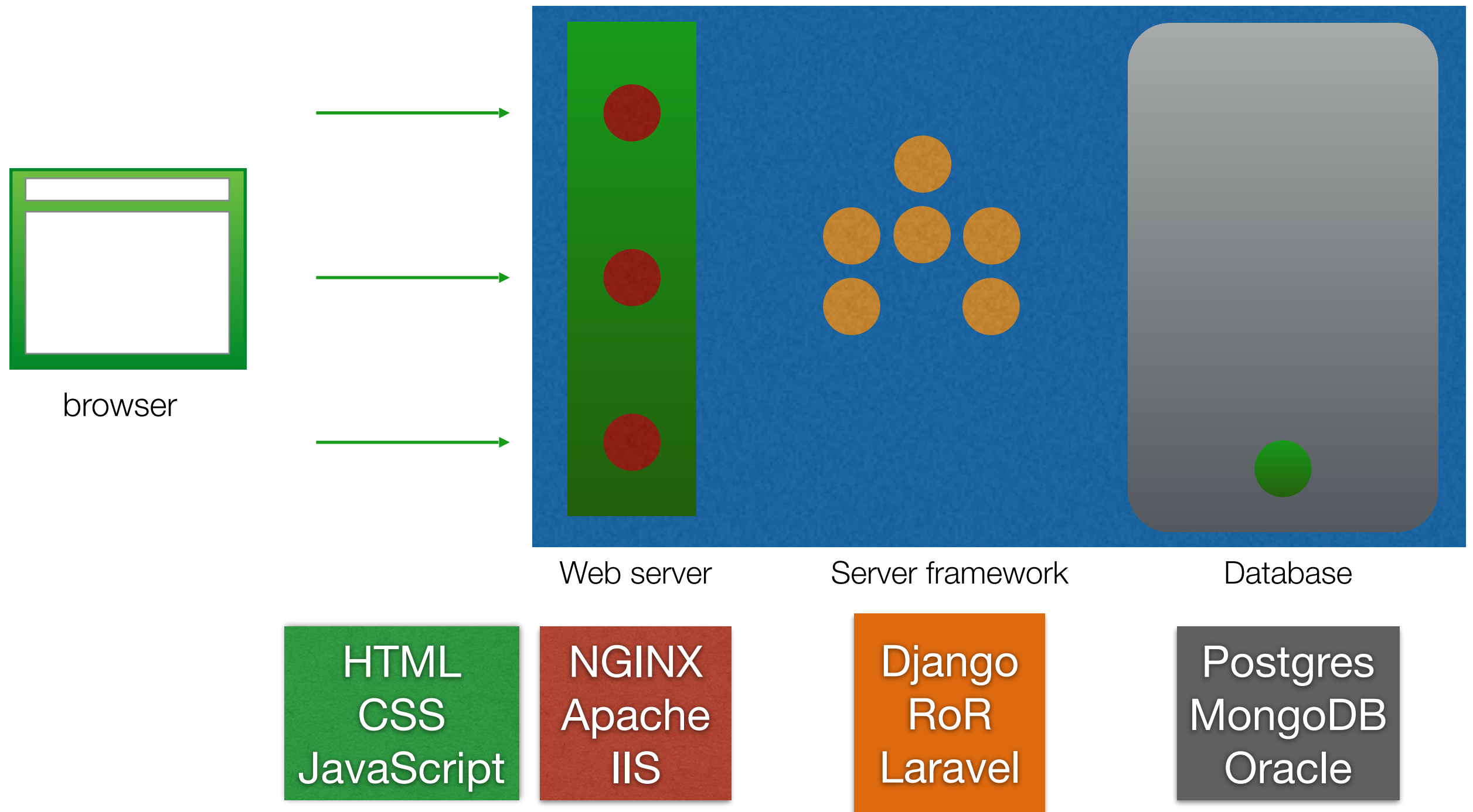
Furniture store



Making the store more responsive

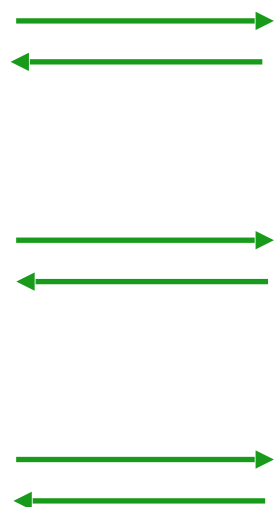


Web development terminology





browser



Web server

Server framework

Database

HTML
CSS
JavaScript

HAML
LESS SASS
CoffeeScript
TypeScript

jQuery
React
Angular
Ember
Vue

NGINX
Apache
IIS
Unicorn
Thin
Puma

Python
Django, Flask

Ruby
Rails, Sinatra

PHP
Laravel, Yii

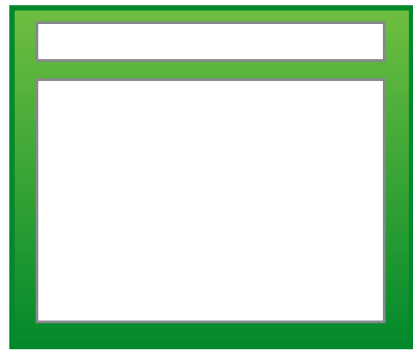
Java
Spring, Play

C#
ASP.NET

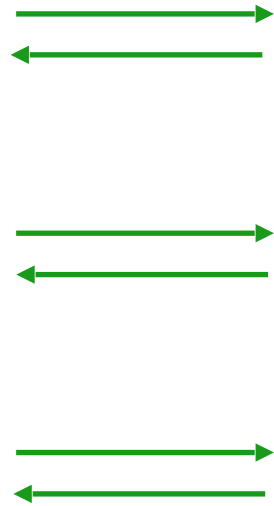
MySQL
Postgres
Microsoft SQL
SQLite
Oracle
MongoDB
Cassandra
Redis
Memcached

Hosting

AWS
Microsoft Azure
Heroku
ps.kz
hoster.kz



browser



Hosting

AWS
Microsoft Azure
Heroku
ps.kz
hoster.kz

Web server

Server framework

Database

HTML
CSS
JavaScript

HAML
LESS SASS
CoffeeScript
TypeScript

jQuery
React
Angular
Ember
Vue

NGINX
Apache
IIS
Unicorn
Thin
Puma

Python
Django, Flask

Ruby
Rails, Sinatra

PHP
Laravel, Yii

Java
Spring, Play

C#
ASP.NET

MySQL
Postgres
Microsoft SQL
SQLite
Oracle
MongoDB
Cassandra
Redis
Memcached

front end

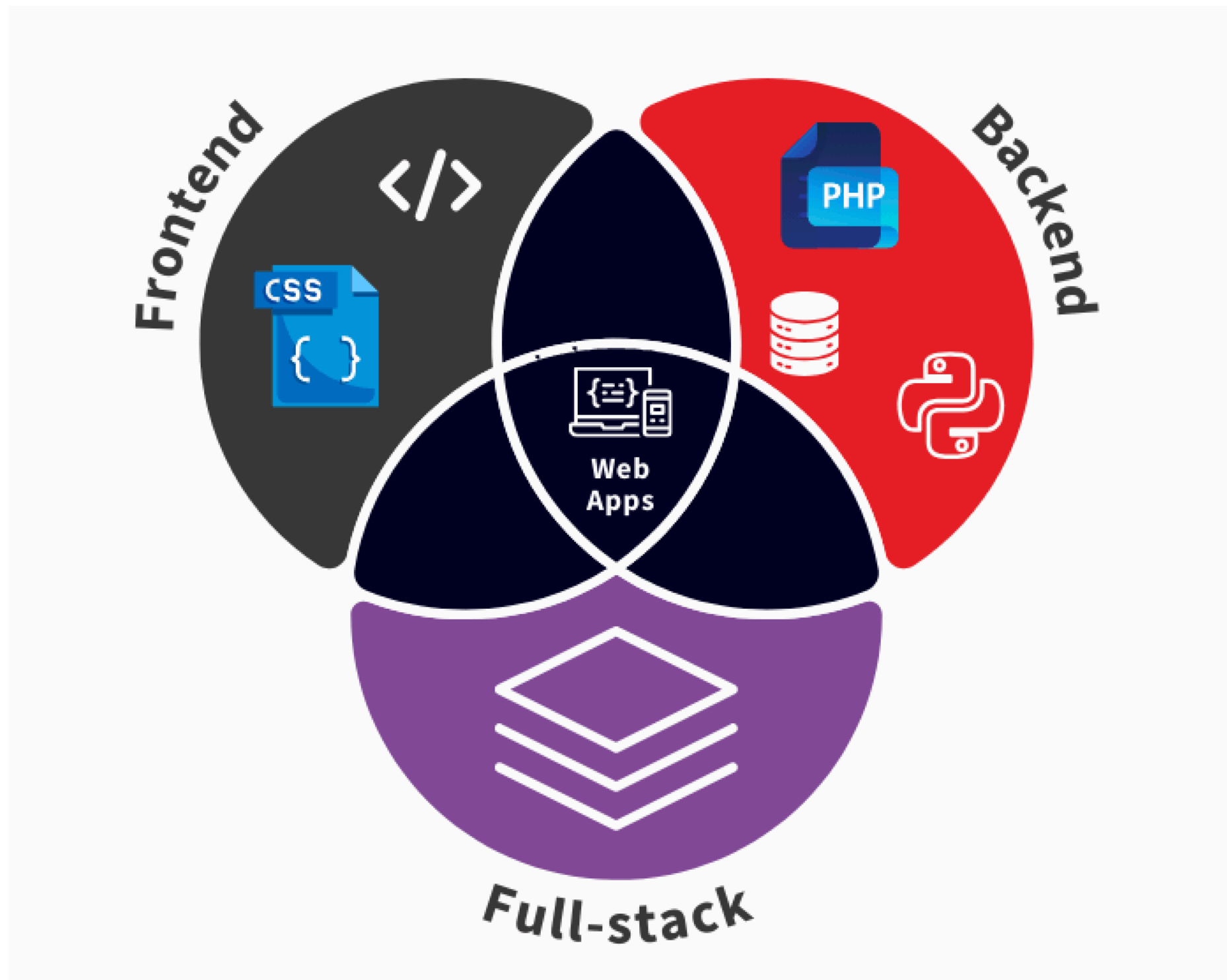
back end

server

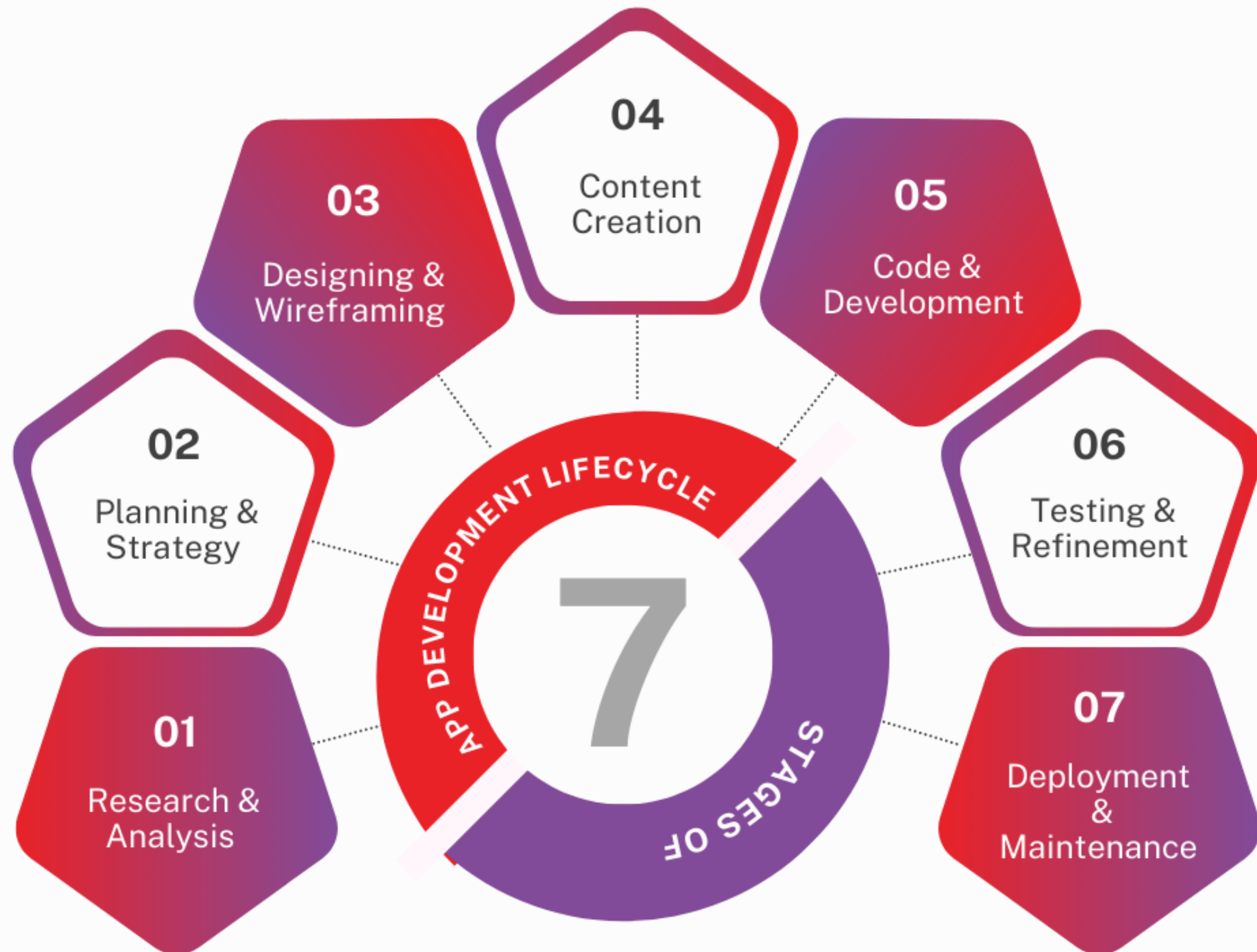
client side

server side

What is Web Application Development?



What is Web Application Development?



Web Application Development Methodologies

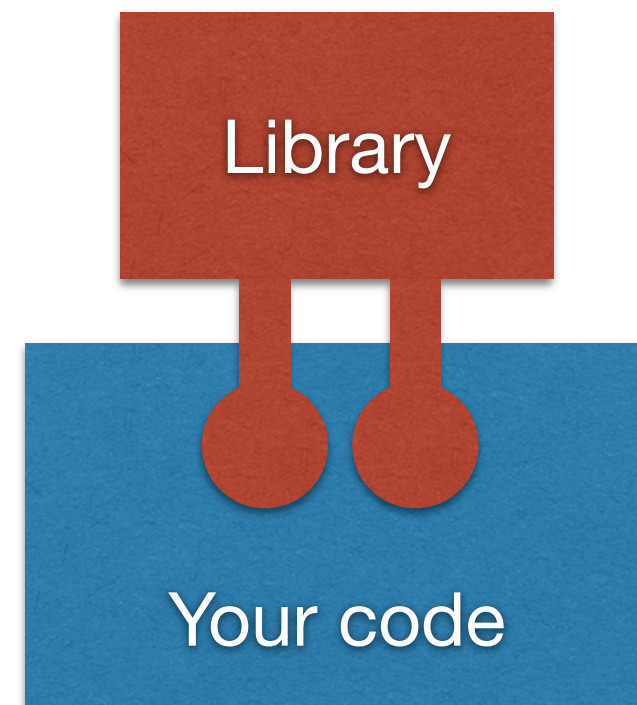
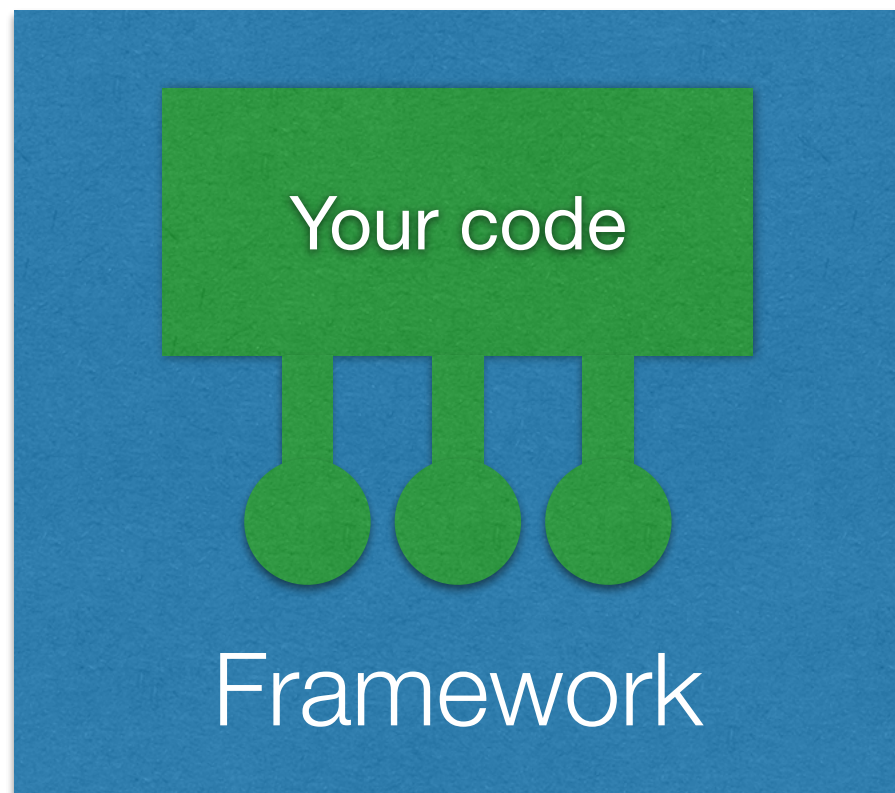
1. **Waterfall:** A sequential process where each phase (requirements, design, development, testing) is completed before moving to the next. Best for fixed, well-defined projects.
2. **Agile:** An iterative and flexible approach emphasizing collaboration, regular feedback, and incremental deliveries. Ideal for evolving requirements.
3. **Scrum:** A framework within Agile using sprints (short, time-boxed iterations) and daily meetings to ensure progress and adaptability.
4. **Spiral:** Combines iterative and waterfall methods, emphasizing risk analysis and gradual refinement. Suitable for high-risk projects.
6. ...

Technologies

Frontend technologies

	Angular	React	Vue.js
Type	Framework	Library	Framework
Language	TypeScript	JavaScript/TS	JavaScript/TS
Performance	High	High	High
Size	Large (~500 KB)	Medium (~100 KB)	Small (~60 KB)
State Management	Built-in (RxJS)	External (Redux)	Built-in (Vuex)
Use Cases	Enterprise apps	Dynamic UIs	Rapid development

Framework & Library



Framework & Library

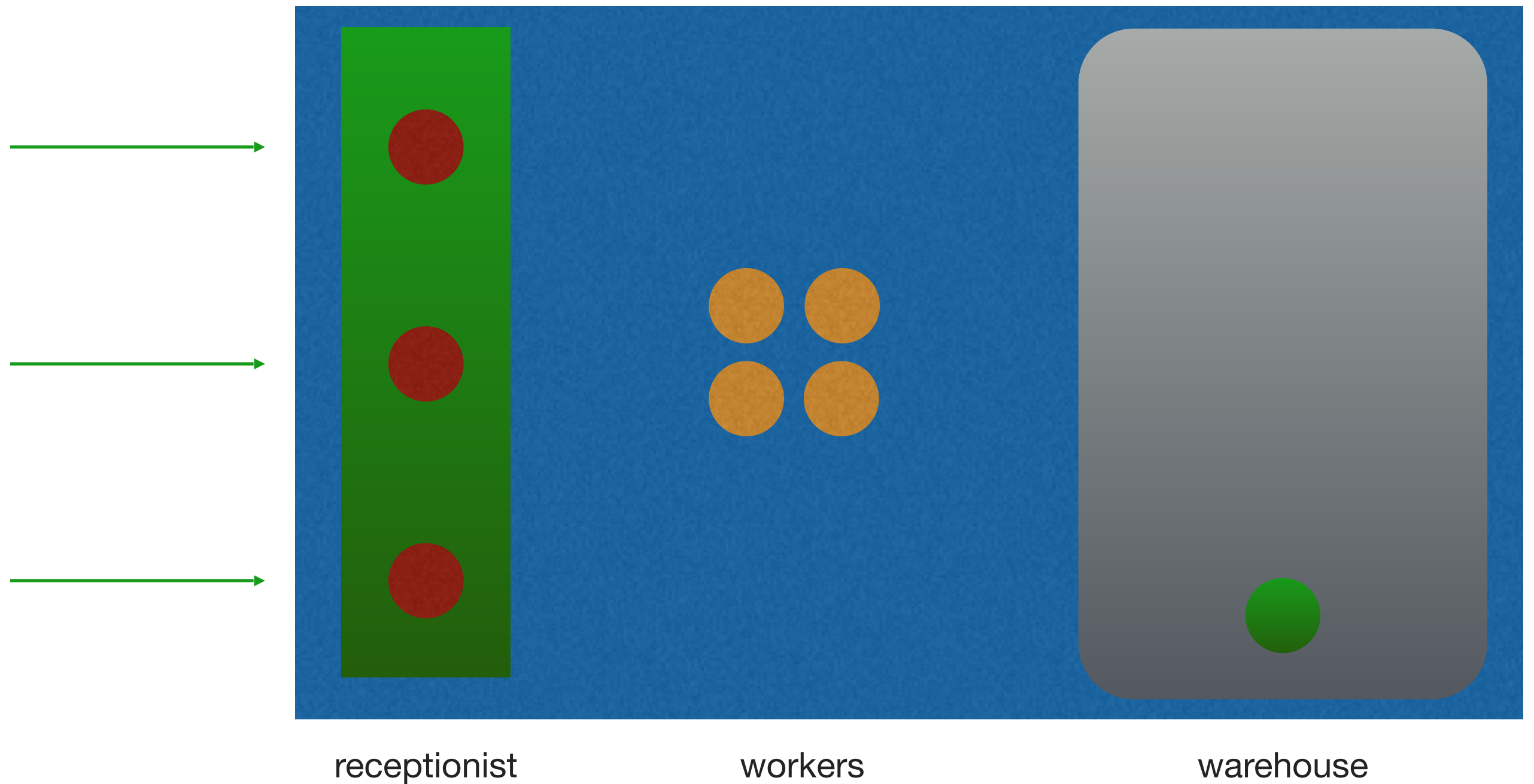
	Library	Framework
Control	You call it.	It controls flow.
Flexibility	More flexible.	Less flexible.
Size	Smaller, task-focused.	Larger, all-in-one.
Learning Curve	Easier to learn.	Steeper learning curve.
Examples	React, Lodash, jQuery	Angular, Django, Rails

Backend technologies

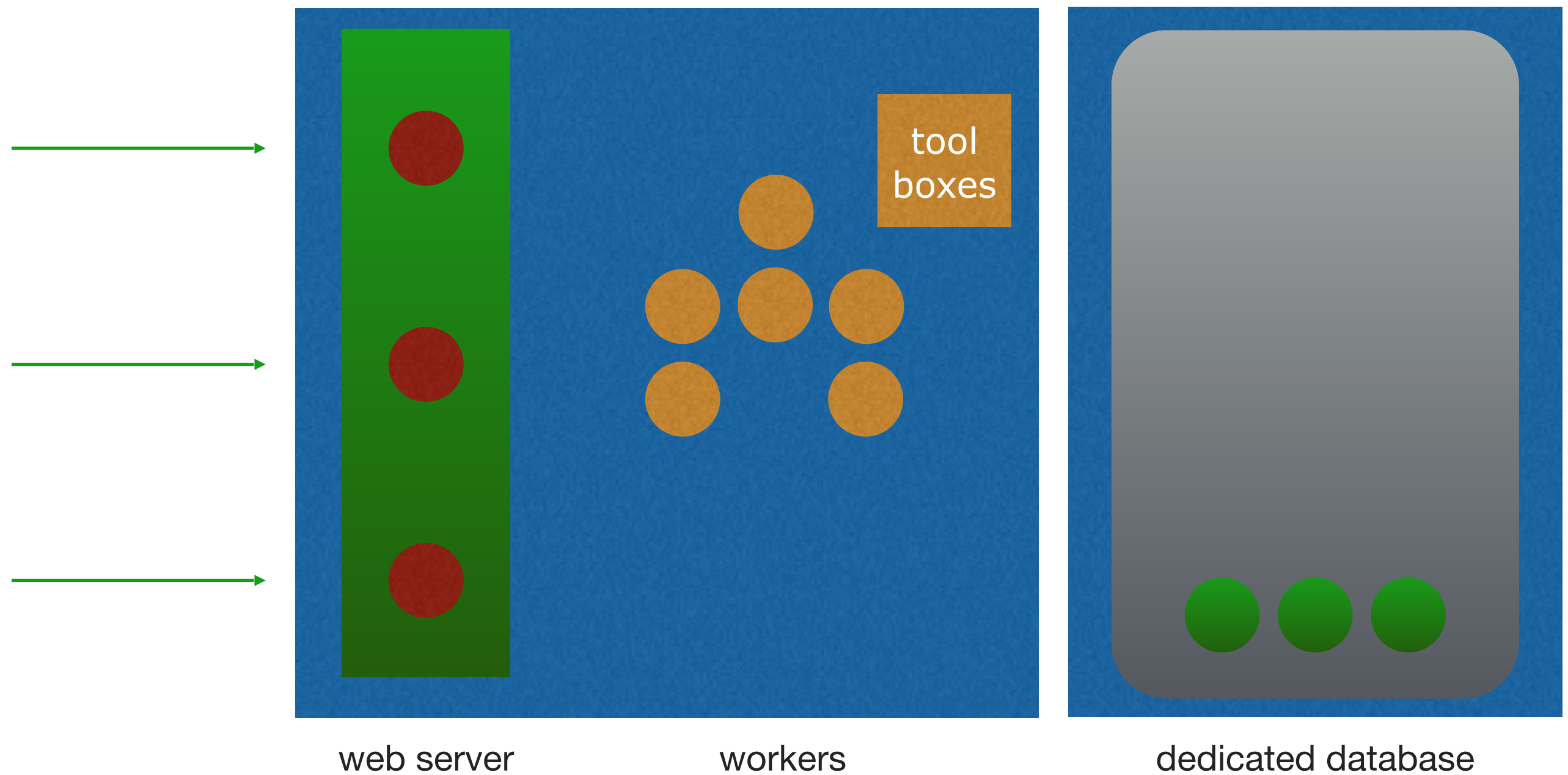
	Django	Flask	Express.js	Ruby on Rails
Type	Full-stack framework	Micro-framework	Minimalist framework	Full-stack framework
Language	Python	Python	JavaScript (Node.js)	Ruby
Performance	High (built-in tools)	High (lightweight)	Very High	High
Flexibility	Flexible	Very flexible	Very flexible	Less flexible
Use Cases	REST APIs, large apps	Prototypes, small apps	APIs, real-time apps	E-commerce, MVPs
Examples	Instagram, Disqus	Pinterest, LinkedIn	Uber, Netflix	Shopify, GitHub

Scaling concepts

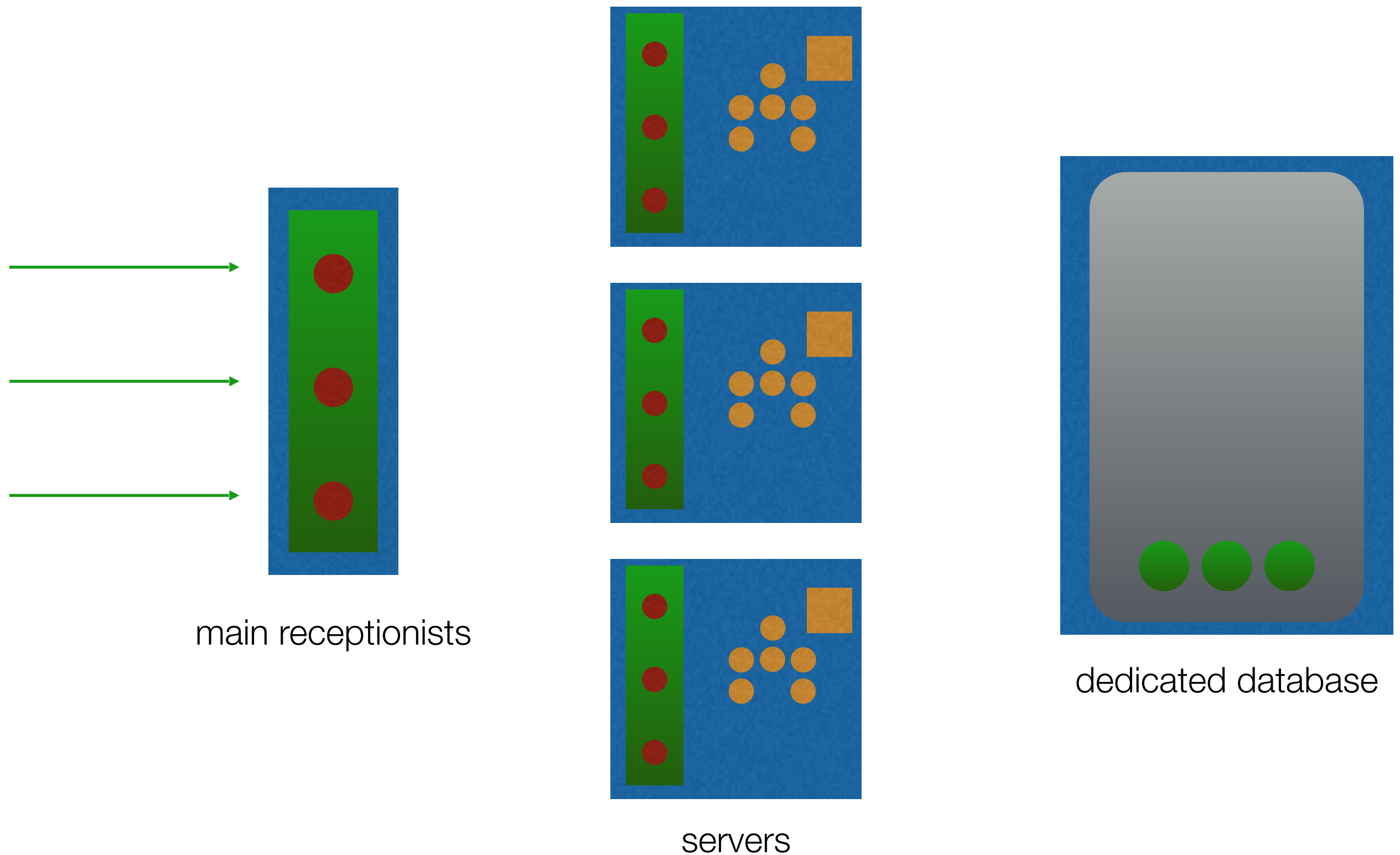
Furniture store



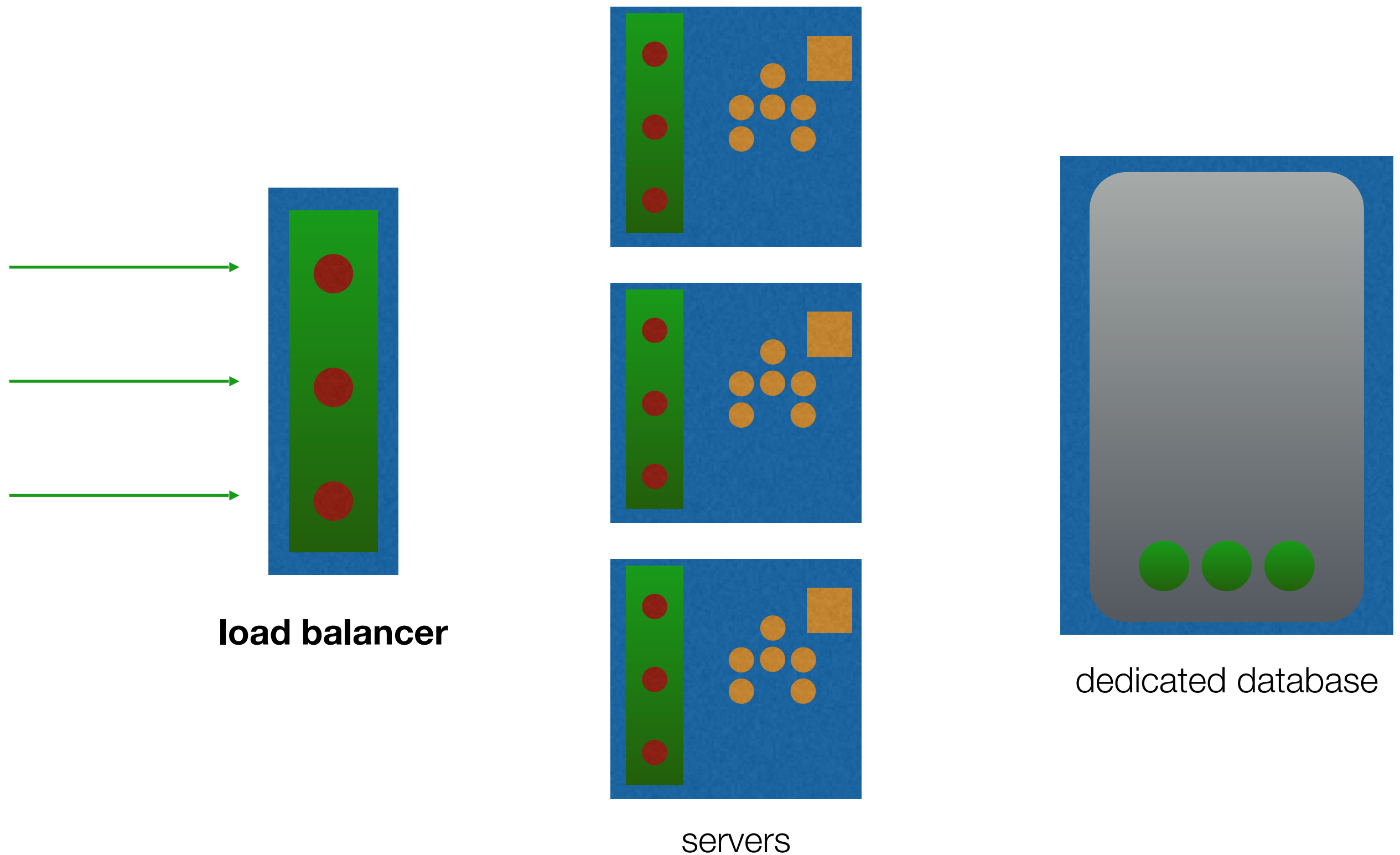
Separating the data store



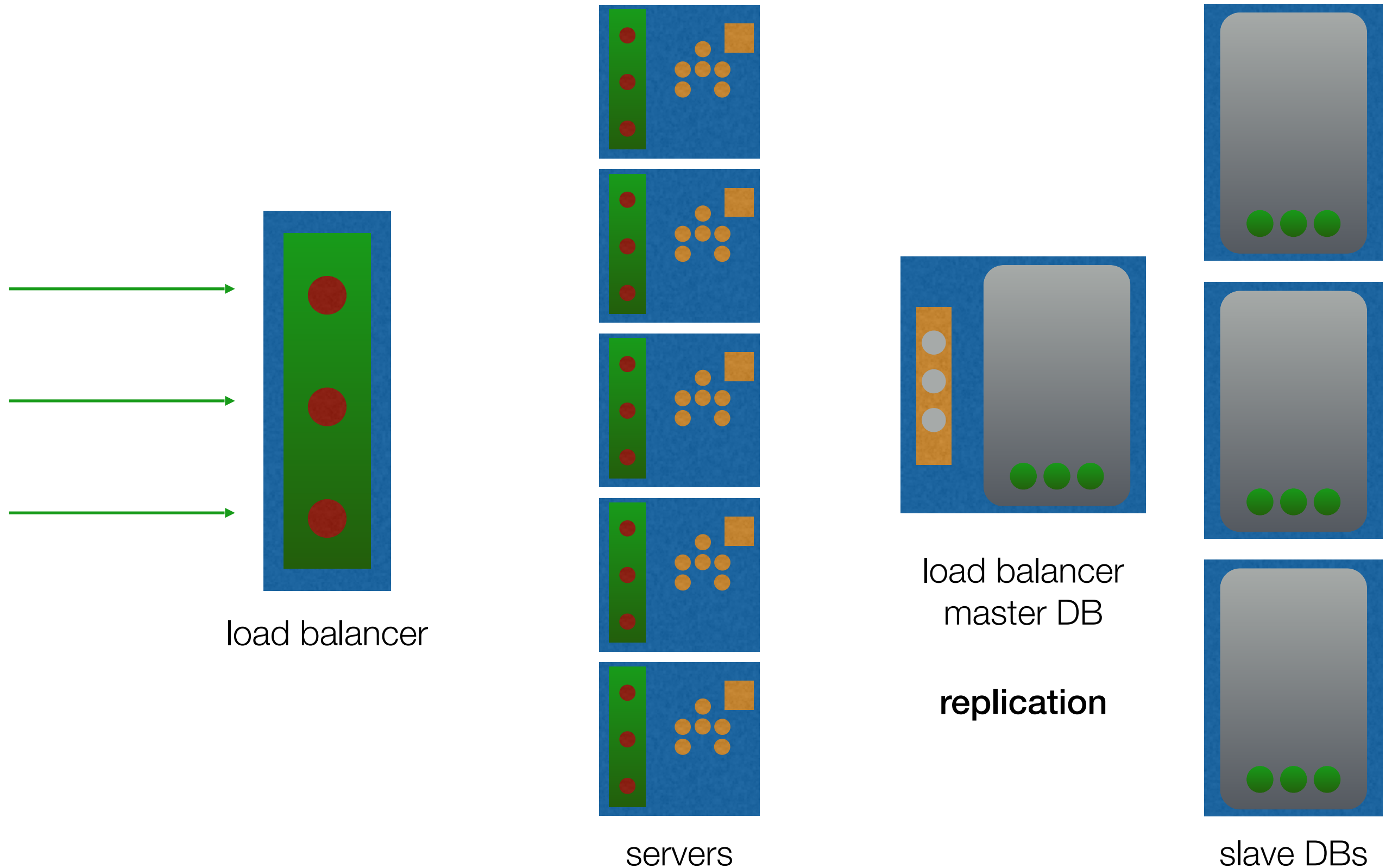
Scaling the server



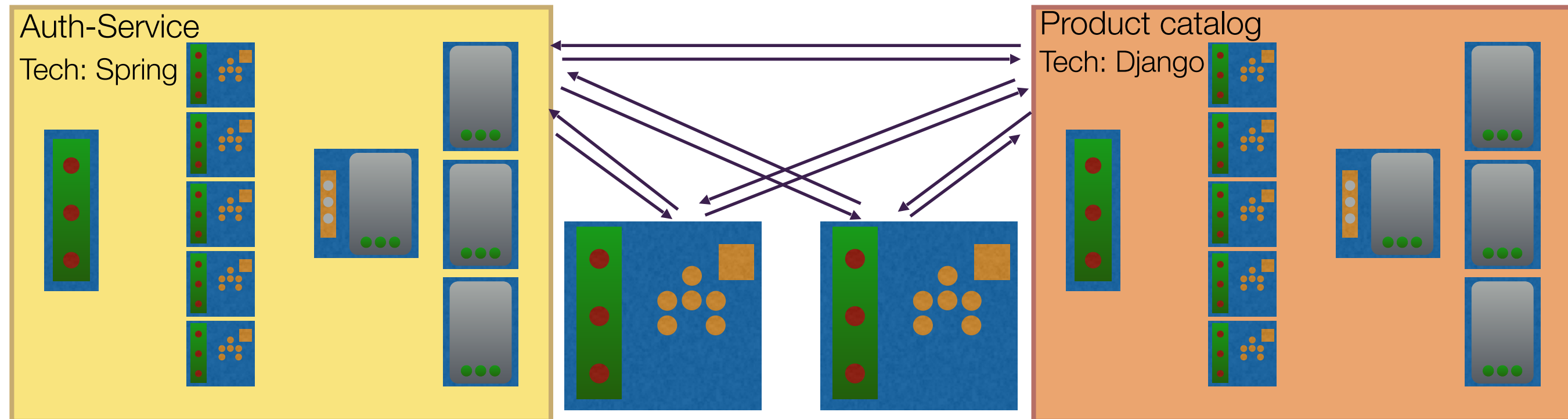
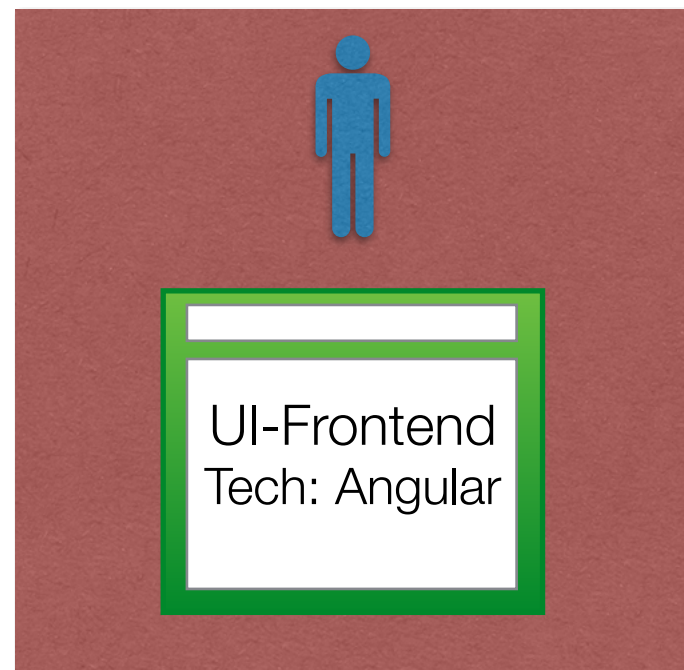
Scaling the server



Scaling the data store



Modularization and APIs



What is API?



API — is like an artist performing on stage, and its users are the audience



RESTful API

1. REST (REpresentational State Transfer) — is an architectural style for developing web services
2. API (Application Program Interface) — is code that allows two software programs to communicate with each other

API endpoint for Companies

1. `/getAllCompanies`
2. `/addNewCompany`
3. `/showCompanyDetail?id=23`
4. `/deleteCompany?id=23`



The URL is a sentence, where resources are nouns and HTTP methods are verbs.

1. `/companies` (GET)
2. `/companies` (POST)
3. `/companies/23` (GET)
4. `/companies/23` (DELETE)



Data formats

- JSON

```
{  
  "root": {  
    "age": "18",  
    "isStudent": "true",  
    "name": "Nick"  
  }  
}
```

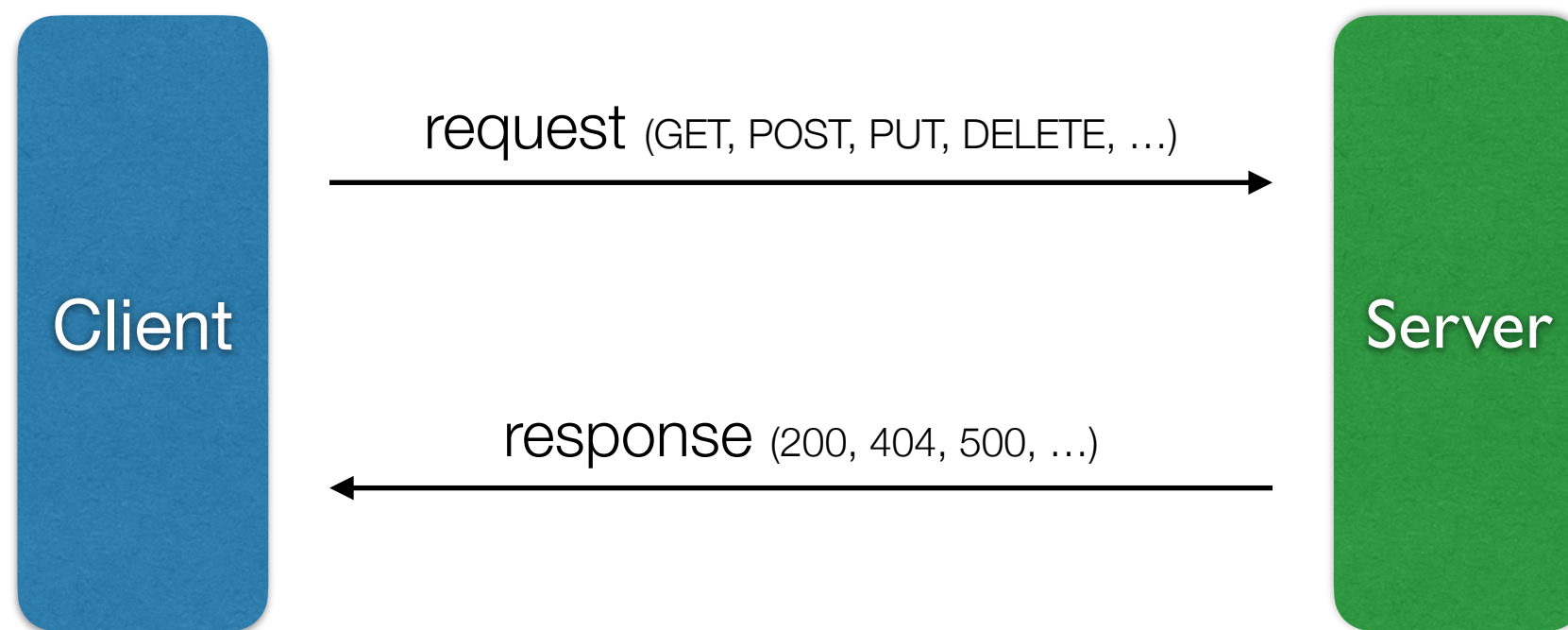
- XML

```
<?xml version="1.0" encoding="UTF-8"?>  
<root>  
  <age>18</age>  
  <isStudent>true</isStudent>  
  <name>Nick</name>  
</root>
```

- CSV

```
name,age,isStudent  
Nick,18,true
```


Client Server Communication



Protocols

- TCP/IP — Transmission Control Protocol / Internet Protocol
 - communication among computers on Internet
- HTTP — Hyper Text Transfer Protocol
 - Communicates with browsers to send web page packets
- HTTPS — Hyper Text Transfer Protocol Secure
 - HTTP with Secure Sockets Layer (SSL)
- FTP — File Transfer Protocol
 - Used by FTP Clients to transfer file packets

HTTP response status codes

- 2xx — Success category
 - 200 Ok
 - 201 Created
- 3xx — Redirection Category
 - 304 Not Modified
- 4xx — Client Error Category
 - 400 Bad Request
 - 401 Unauthorized
 - 403 Forbidden
 - 404 Not Found
- 5xx — Server Error Category
 - 500 Internal Server Error
 - 503 Service Unavailable

Questions?