Calling services via proxy

Two potential setups for prod:

- Services and Front End always on diff servers
    - Ex: `example.com` and `api.example.com`
    - We won't do this, but is also common
- Services and Front End on same servers in prod
    - Both easier AND harder in dev
    - Can avoid server domains in fetch()
        - Easy to swap between dev and prod
    - You want just paths for fetch()
        - Ex: `/api/cats`
        - Not: `https://api.example.com/api/cats`

Dev Server doesn't support services

The CRA dev server doesn't let us write services

- So we would need a domain+port in `fetch()`
- But those urls would change in prod
    - Never good to change code to deploy
        - Mistakes not seen in development

Solution: Proxy the service calls

CRA Dev Server:

- Has front end files
- Front end calls services as path
- If a service call comes in
    - **Proxy** it to services server
    - Return results

Dev services server:

- Configured for static files
    - But doesn't have any
- Has services

Moving to Prod

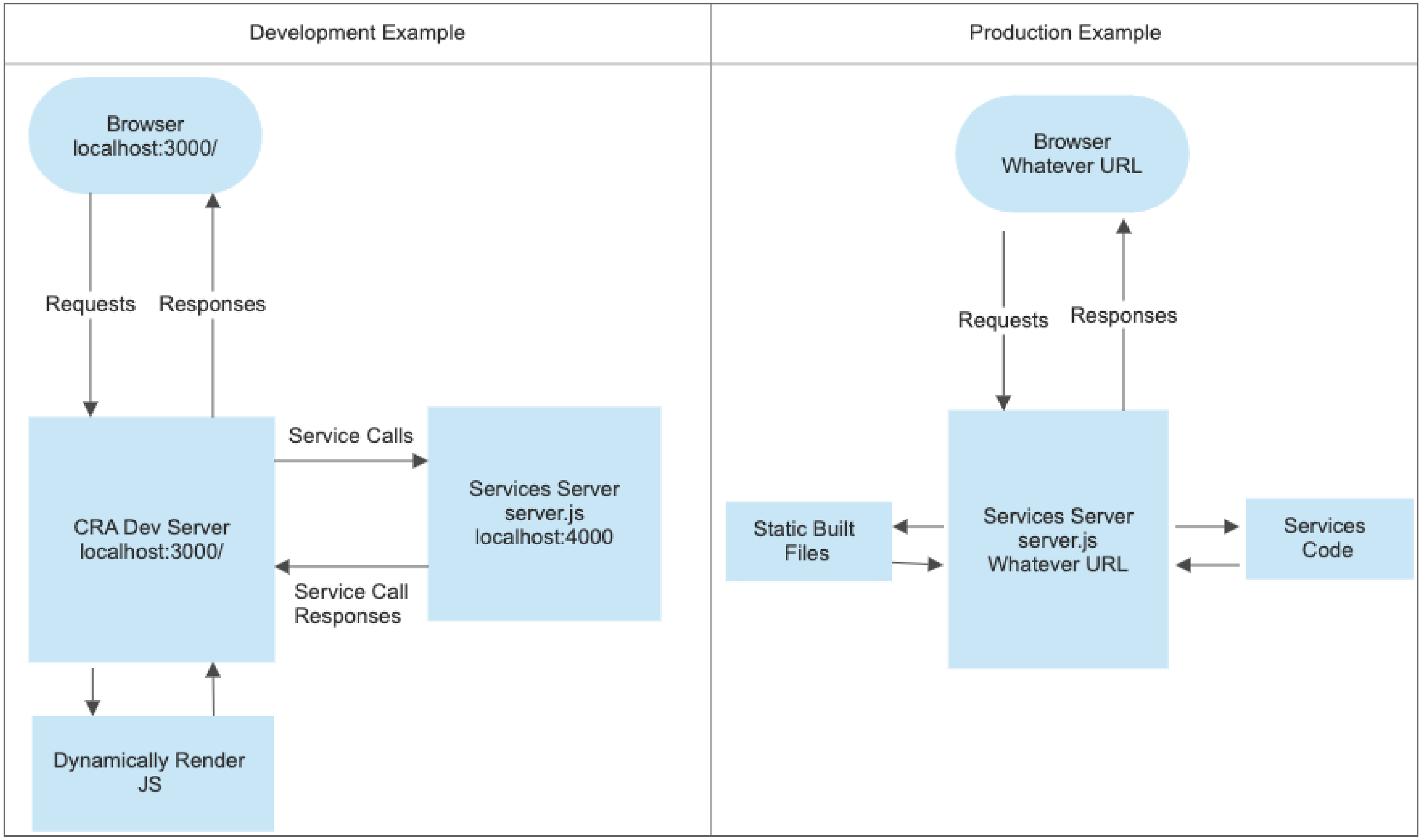
Once a round of development is done:

Prod Server:

- Same server code as dev services server
- Given static files from CRA `npm run build`
- Service calls made from front end are to paths
- Instead of proxying
    - Responds to service calls itself

No changes in code between dev and prod

Make sure this makes sense!



CRA Dev Server

The CRA Dev Server is only for development

- This is what `npm start` runs by default
  - You can change that default
  - `react-scripts start`

Use in Development as normal

- We will configure the `proxy`
- Run `npm run build` to create static built files

This server is used only in dev

Services Server Setup

Serves both static files and services

- Build express server (server.js)
- Configure `express.static()`
    - Hint: `build/` instead of `public/` is convenient
- Write routes for service endpoints
- Set the PORT to be different than CRA dev server
    - Hint: set PORT to `process.env.PORT || 3000`
    - Lets us override at command line
        - `PORT=4000 node server.js`

This server used in both dev and prod

Setting up Project

Some teams will have different projects

- Different directories/`package.json` files
- One for server code
- One for client React code
- Great for different teams

Other projects will have just one project

- Which is what we will do for this course
- End result is just one server and files
- Everything coded by us

Creating our server

- Create `server.js` in react project directory
- Install so deps are added to existing `package.json`
    - Ex: `npm install express` in react project
- You can modify the `scripts` in `package.json`
    - `npm run build` stays unchanged
    - `npm run dev` to run current `npm start`
    - `npm start` starts the server.js
- Set `express.static()` to `./build`
    - Serve static files from `npm run build`

Configuring the Proxy

CRA Dev server can proxy service calls in dev

- In prod there will be no proxy
- CRA Dev Server checks the `package.json` for config

Add to `package.json`:

```
"proxy": "http://localhost:4000",
```

- 4000 can be any available port
- 3000 is used by CRA dev server

What does the proxy setting do?

When CRA Dev Server has request it would 404 on

- It (not browser!) will request url from proxy
- Sends response to browser

Browser only talks to CRA Dev Server

- CRA Dev Server is "proxy"
    - To Services Server (server.js)

All `fetch()` urls are only paths

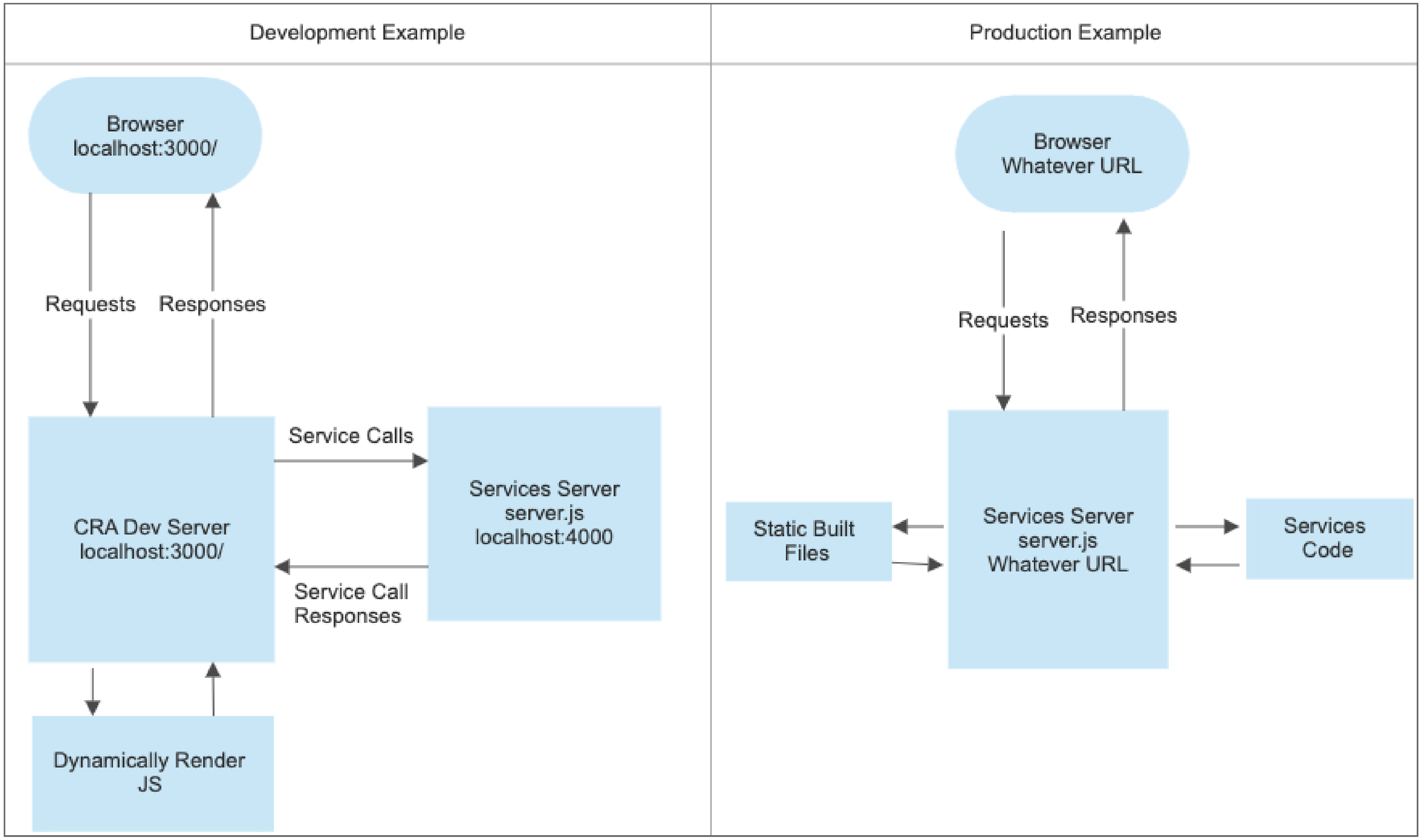
- Same domain+port as page url
- No CORS!

Trying it out

Notice dev and prod will use different ports

- Common to set PORT by environment variable
- In services server:
  - `const PORT = process.env.PORT || 4000;`
- To start services server in dev:
  - `PORT=4000 npm run dev`
- To start server in prod:
  - `PORT=3000 npm start`
  - (Real production would be PORT 443)

Is anything unclear now?



Summary - Why we need a proxy

Want Service Calls URLs in dev to match prod URLs

If Prod services on diff server from pages

- Different solution involved
- But solving same URL problem

When Prod services on same serve as pages

- Proxy with CRA Dev Server
- What we are doing

Summary - What does the Proxy do?

A Service request comes to CRA Dev Server

- It makes same request of Proxy target
- Gets response, sends back to browser

Browser always talking to CRA Dev Server

- `fetch()` URLs point to same server as pages

In Prod

- No Proxy involved, No CRA Dev Server
- Only one server
- Has both built static files and services

Summary - Services server

Just one way to do it, no single "best"

- Create an express `server.js`
    - Same `package.json` as react project
- Set `express.static()` to `./build`
- Set PORT to be set by environment
- Write services as normal
- Update `package.json` scripts
    - start services server
    - set PORT for prod vs dev

Summary - Configure Proxy

- Define `proxy` in `package.json`
  - Point to services server url
  - With dev PORT
- `proxy` only impacts CRA Dev Server
  - Not a universal config the way `scripts` is
    - Read specifically by CRA Dev Server
- In Prod proxy setting is ignored
  - No CRA Dev Server to read it
  - Express server doesn't read it