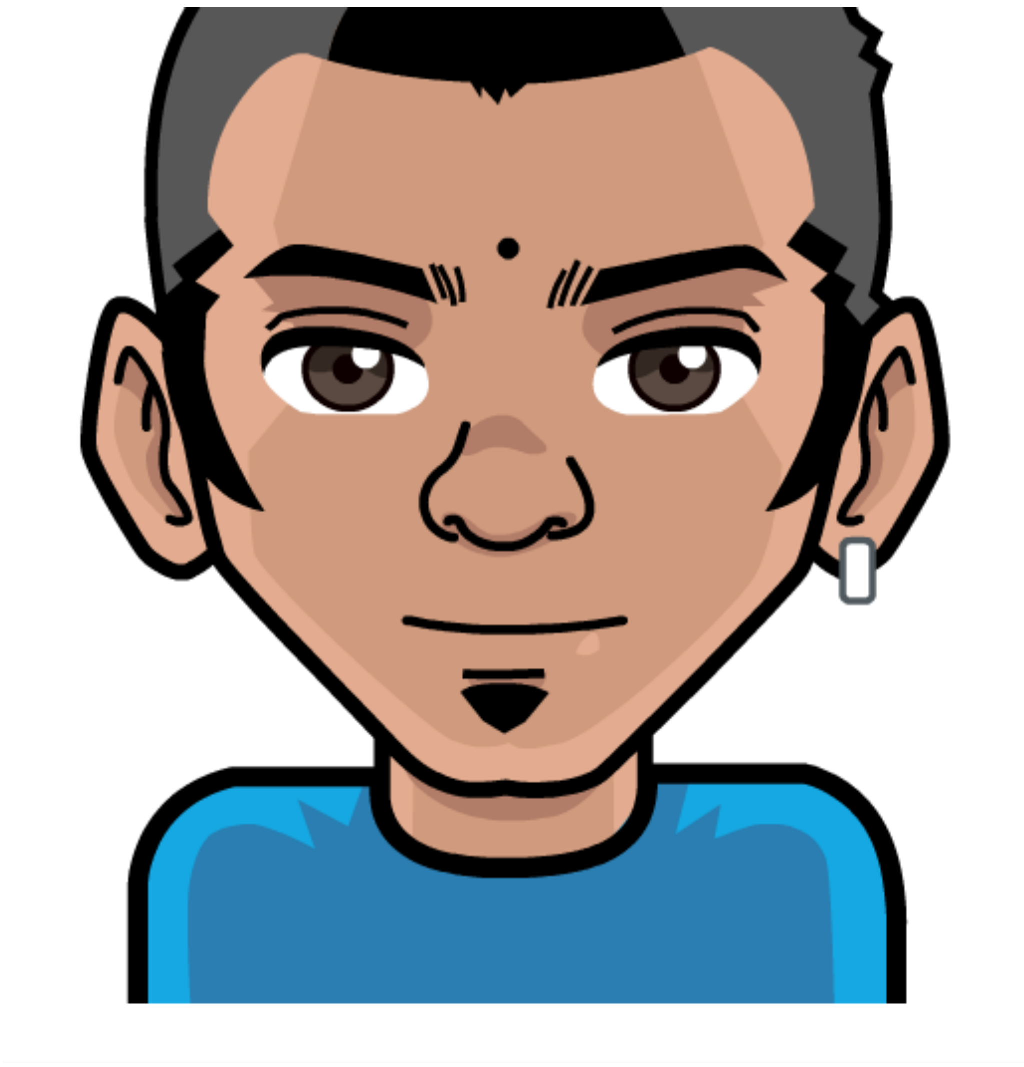
RAJU GANDHI

LEVERAGING THE ANGULAR CLI



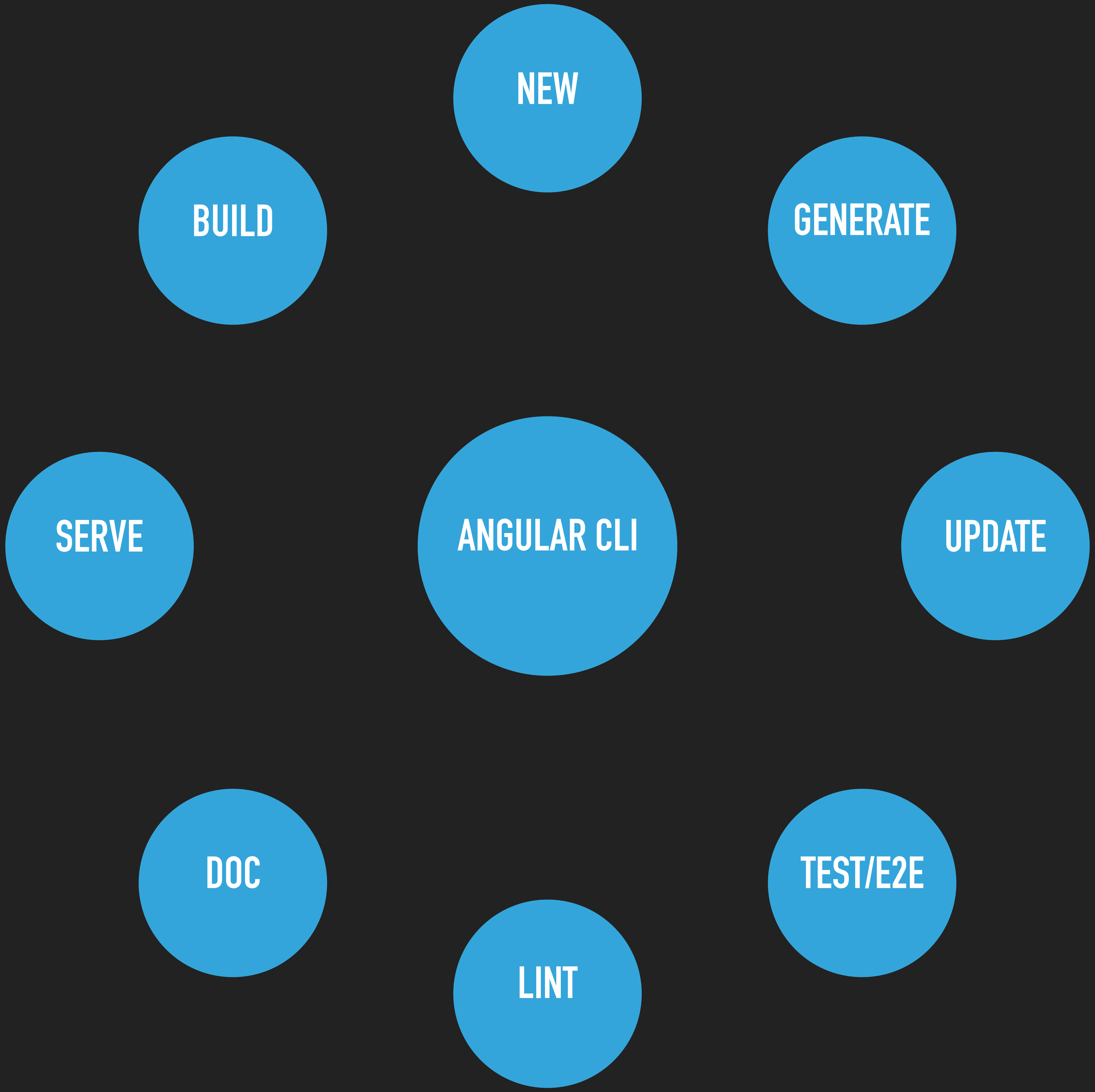
RAJU GANDHI

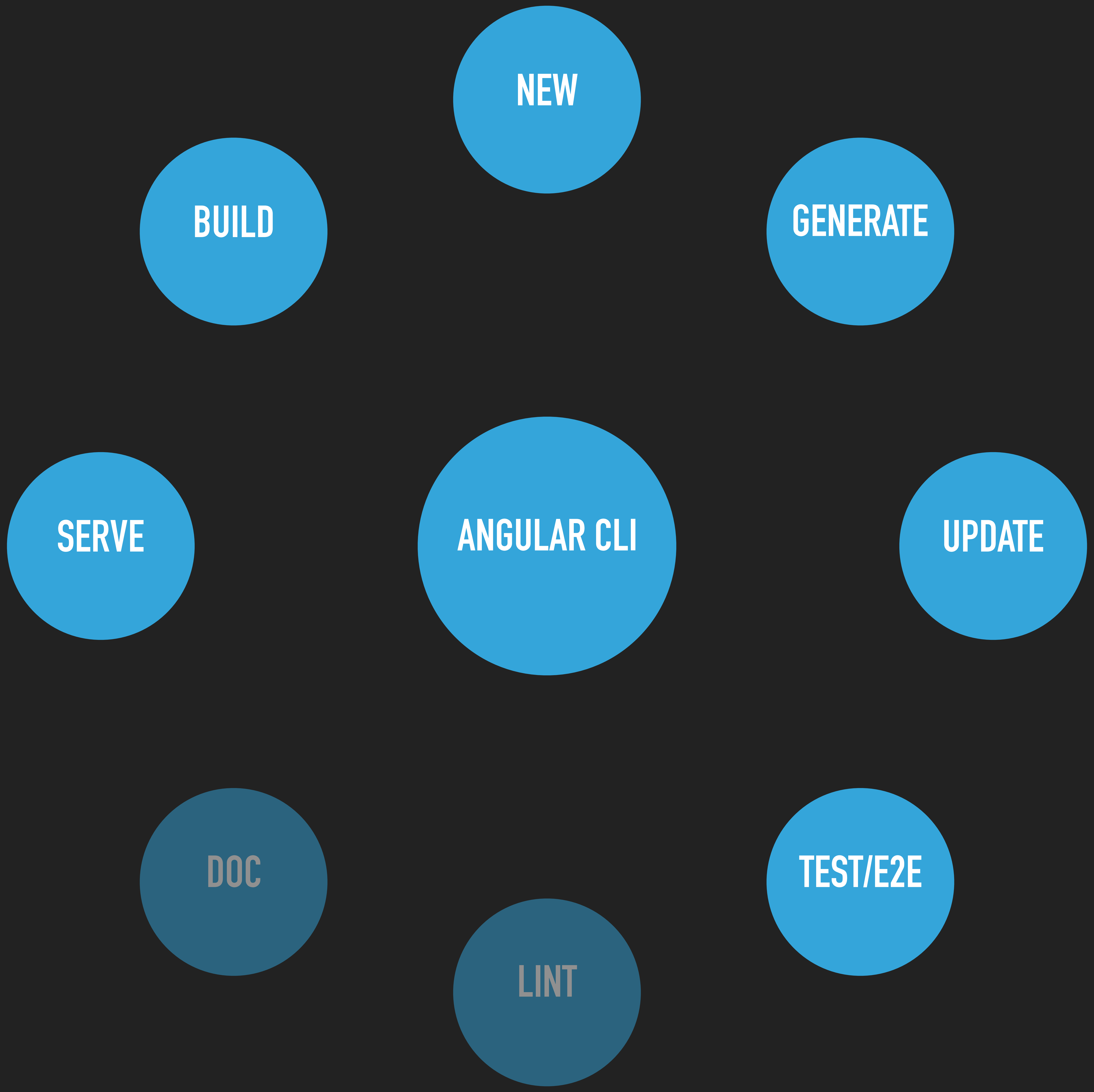
@LOOSELYTYPED

CTO - INTEGRALLIS SOFTWARE

ANGULAR CLI

BUILD TOOLING


NEW
BUILD
GENERATE
SERVE
ANGULAR CLI
UPDATE
DOC
TEST/E2E
LINT



ANGULAR CLI

NEW

GENERATE

UPDATE

TEST/E2E

LINT

DOC

SERVE

BUILD

NPM/YARN

TSLINT

ANGULAR CLI

TSCONFIG

KARMA

SETTINGS

```
ng set --global packageManager=yarn
ng set --global defaults.styleExt=scss
```

```
$ cat .angular-cli.json
{
  "defaults": {
    "styleExt": "scss"
  },
  "packageManager": "yarn"
}
```

NG NEW

NEW

- Generates a new project scaffold
- Provides a `.angular-cli.json` to configure your project

```json
{
  "$schema": "./node_modules/@angular/cli/lib/config/schema.json",
  "project": {
    "name": "some-project"
  },
  "apps": [
    {
      "root": "src",
      // ...
    }
  ],
  "defaults": {
    "styleExt": "scss",
    "component": {}
  }
}
```

```
{
  "$schema": "./node_modules/@angular/cli/lib/config/schema.json",
  "project": {
    "name": "some-project"
  },
  "apps": [
    {
      "root": "src",
      // ...
    }
  ],
  "defaults": {
    "styleExt": "scss",
    "component": {}
  }
}
```

NEW - MULTIPLE APPS

- Can support multiple angular applications with one project
- Each app can have a "name"
  - Use `ng serve --app <index>` or `ng serve --app <name>`
- **Note** that `e2e`, `lint` and `test` are shared!
  - However, you can supply an "app specific" file at the command line
  - `ng e2e --config /path/to/app/specific/config`

```
{
  "$schema": "./node_modules/@angular/cli/lib/config/schema.json",
  "project": {
    "name": "some-project"
  },
  "apps": [
    {
      "root": "src",
      // ...
    }
  ],
  "defaults": {
    "styleExt": "scss",
    "component": {}
  }
}
```

```
{
  "defaults": {
    "styleExt": "css",
    "component": {},
    "build": {
      "sourcemaps": true,
      "baseHref": "/"
    },
    "serve": {
      "port": 4200,
      "proxyConfig": "proxy.config.json"
    }
  }
}
```

NG GENERATE

GENERATE

- Code generation via Schematics
- Supports aliases for almost all commands

```
ng help generate
Available schematics:
    application
    class
    component
    directive
    enum
    guard
    interface
    module
    pipe
    service
    universal
    appShell
```

```
ng help generate
Available schematics:
    application
    class
    component
    directive
    enum
    guard
    interface
    module
    pipe
    service
    universal
    appShell
```

```
> ng g m Core
> # same as
> ng generate module Core
> # produces
  create src/app/core/core.module.ts (188 bytes)
```

```
> ng g s Api -m Core
> # same as
> ng generate service Api --module Core
> # produces
  create src/app/api.service.spec.ts (356 bytes)
  create src/app/api.service.ts (109 bytes)
  update src/app/Core/Core.module.ts (260 bytes)
```

```
ng g cl Todo --type model
  create src/app/todo.model.ts (22 bytes)
```

NG UPDATE

UPDATE

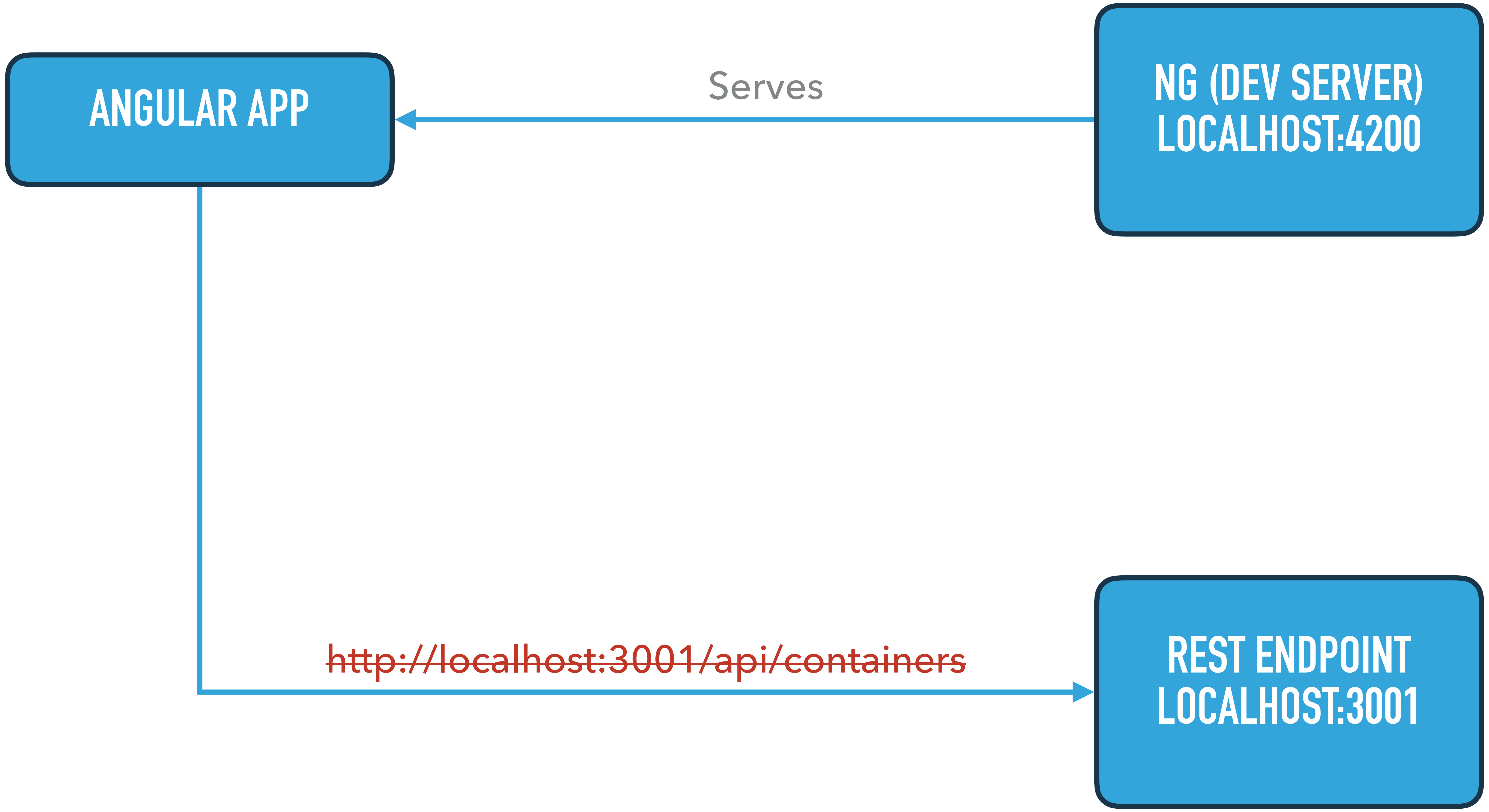
- Only updates package.json
- Use judiciously

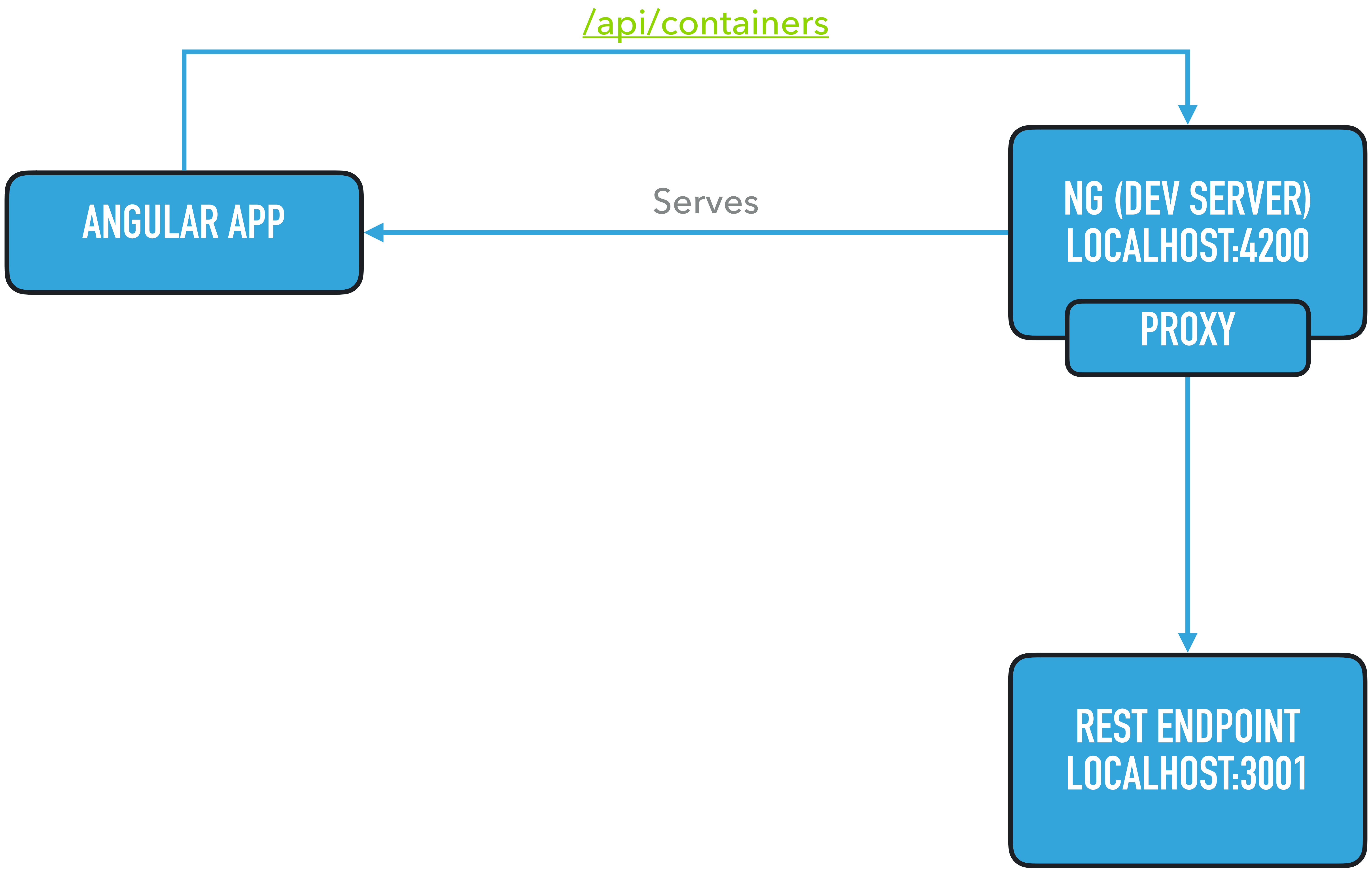
NG TEST

```
$ ng test --browser ChromeHeadless # watch mode
$ ng test --browser ChromeHeadless --single-run # ci
```

NG SERVE

CORS

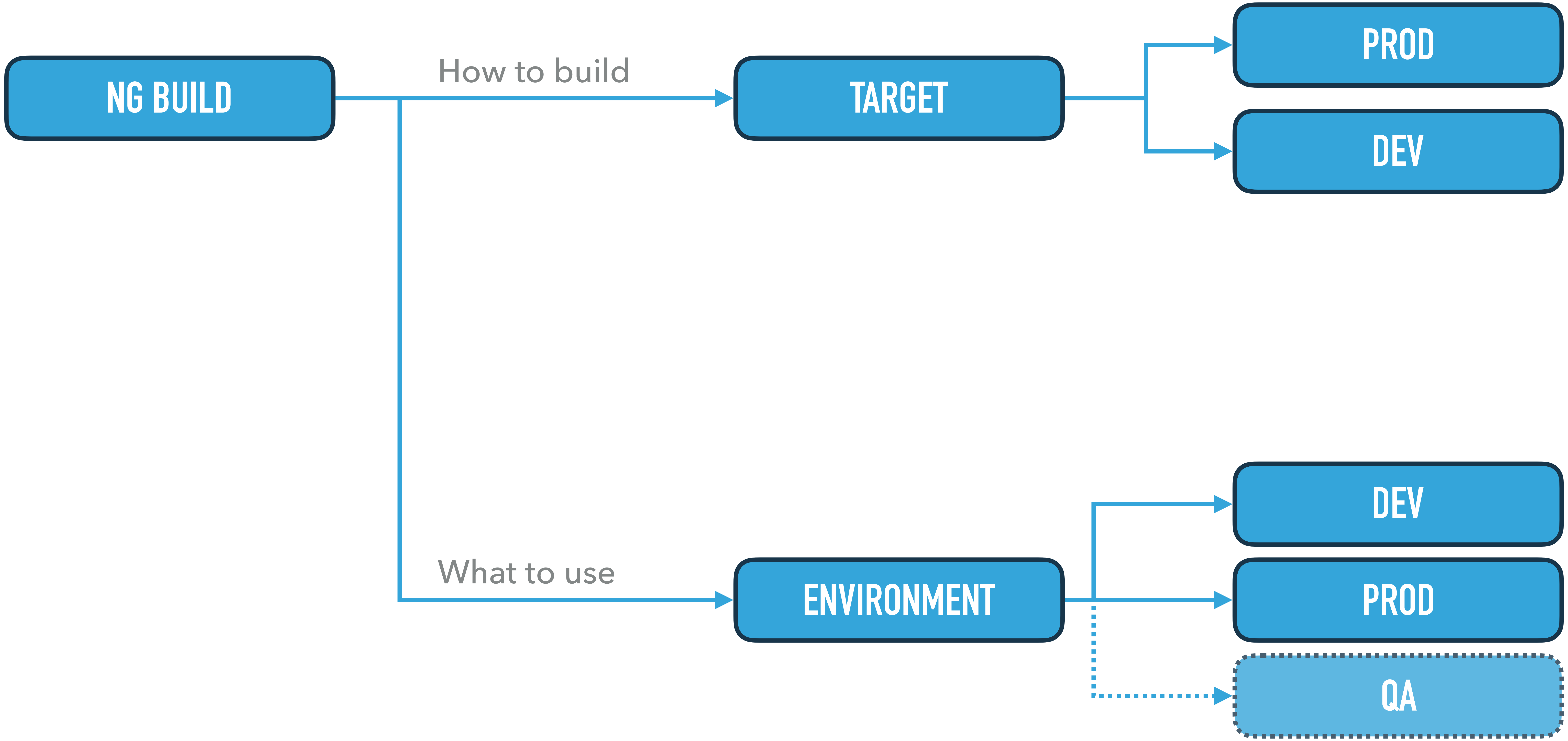

ANGULAR APP
Serves
NG (DEV SERVER)
LOCALHOST:4200
~~http://localhost:3001/api/containers~~
REST ENDPOINT
LOCALHOST:3001


/api/containers
ANGULAR APP
Serves
NG (DEV SERVER)
LOCALHOST:4200
PROXY
REST ENDPOINT
LOCALHOST:3001

```
> cat proxy.config.json
{
  "/api": {
    "target": "http://localhost:3001",
    "secure": false
  }
}
> ng serve -pc proxy.config.json
```

```
# angular service
http.get('/api/containers');
```

NG BUILD

NG BUILD
How to build
TARGET
PROD
DEV
What to use
ENVIRONMENT
DEV
PROD
QA

```
import {
  isDevMode
} from '@angular/core';

import {
  environment
} from '../environments/environment';
```

BUILD

- `dev` / `prod` are the only “targets” supported
  - These are “meta” flags that affect compilation (JIT vs AOT), sourcemaps etc
- You can add as many other “environments” as you like
  - Create `src/environments/environment.NAME.ts`
  - Update `.angular-cli.json` with the same `NAME`
  - Invoke with `ng build --env NAME`

EXTERNAL TOOLS

```
> # install compodoc globally
> npm install -g @compodoc/compodoc
> # generate docs
> compodoc -p ./tsconfig.json
```

COMPODOCS

- Supports
  - Visualizing your modules/components/classes/enums/services
  - Test coverage
  - Themes
  - Search
  - watch and serve

```
> # install webpack-bundle-analyzer
> npm install --save-dev webpack-bundle-analyzer

> # generate stats file in dist/stats.json
> ng build --prod --env=prod --stats-json

> webpack-bundle-analyzer dist/stats.json
```

ALL DONE

THANKS!!

RAJU GANDHI