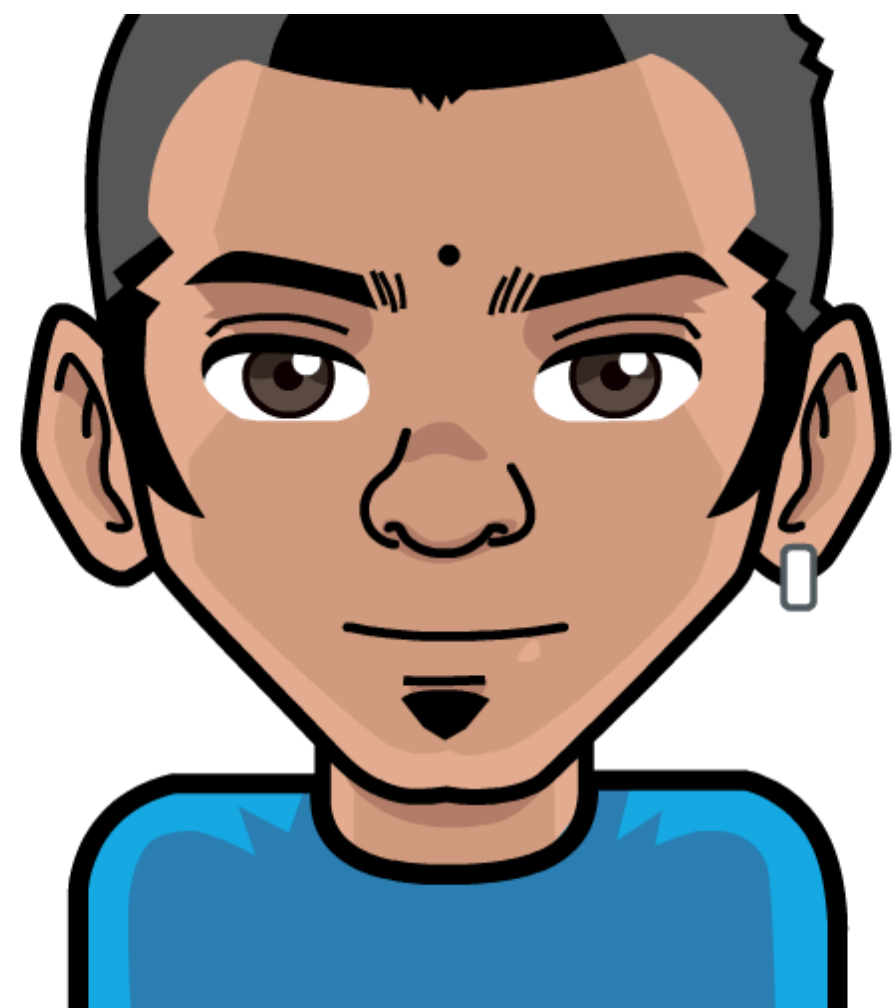





RAJU GANDHI

---

# LEVERAGING THE ANGULAR CLI

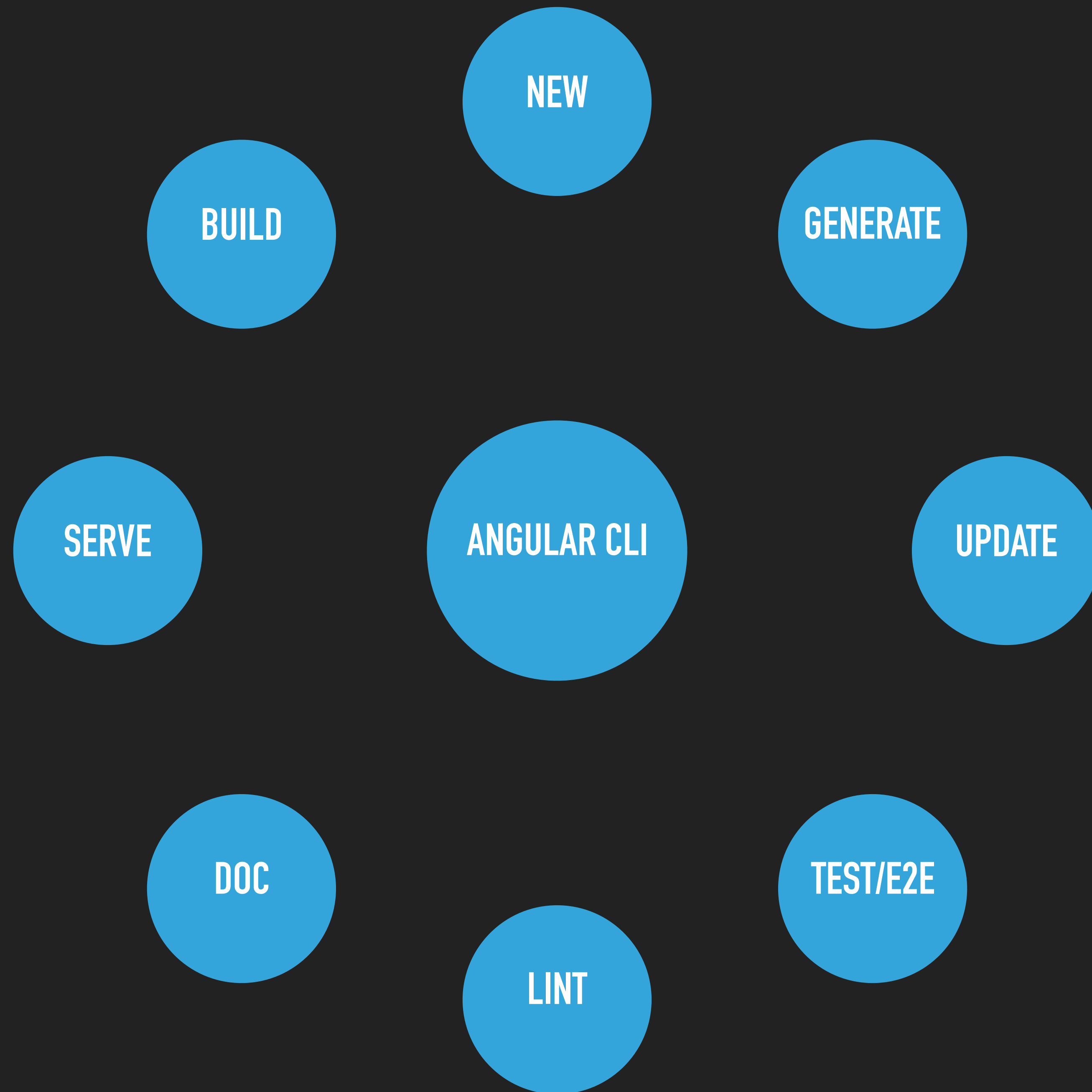


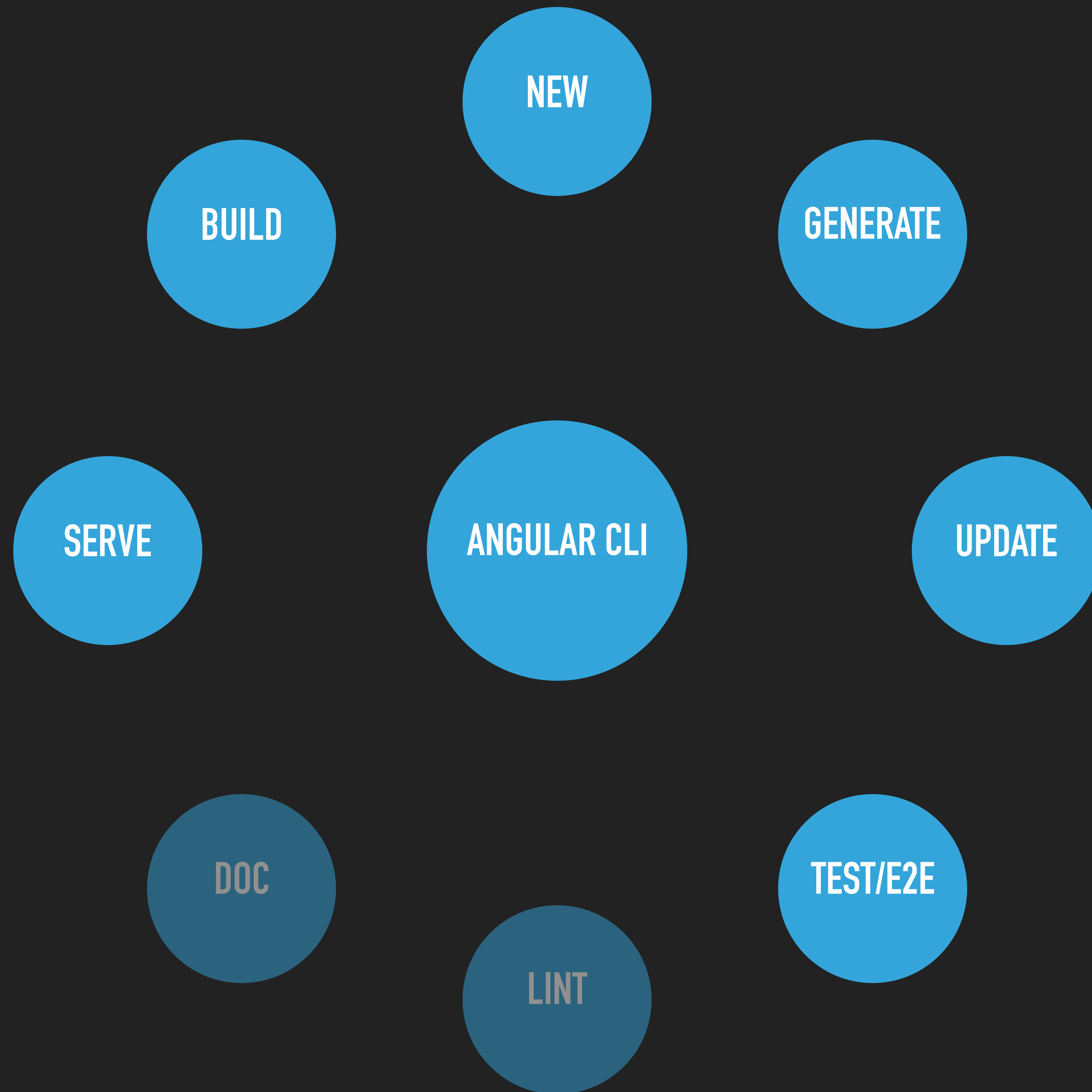
# RAJU GANDHI

   @LOOSELYTYPED  
CTO - INTEGRALLIS SOFTWARE

# ANGULAR CLI

**BUILD TOOLING**





```
graph TD; A(( )) --- B((NPM/YARN)); A --- C((TSLINT)); A --- D((TSCONFIG)); A --- E((KARMA)); A --- F((ANGULAR CLI));
```

**NPM/YARN**

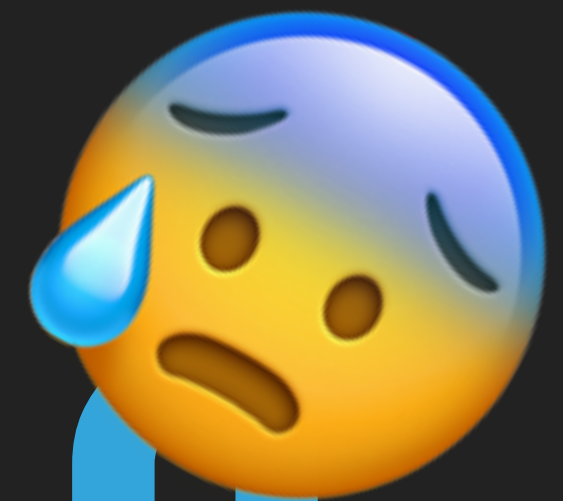
**TSLINT**

**ANGULAR CLI**

**TSCONFIG**

**KARMA**

1.7.4 -> 6.0.0





# SETTINGS

```
ng set --global packageManager=yarn
```

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

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

```
ng config -g cli.packageManager yarn
```

```
$ cat ~/angular.json
{
  "version": 1,
  "cli": {
    "packageManager": "yarn"
  },
}
```

<https://github.com/angular/devkit/tree/master/packages/schematics/angular>

NG NEW

### NEW

- ▶ Generates a new project scaffold

NEW

1.7.4

- ▶ Provides a `.angular-cli.json` to configure your project



```
{
  "$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

1.7.4

- ▶ 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"
    }
  }
}
```

NEW

6.0.x

- ▶ Provides a **angular.json** to configure your project
- ▶ To migrate from 1.7.x project
  - ▶ `npm i --save-dev @angular/cli@6.0.0`
  - ▶ `ng update @angular/cli --migrate-only --from=1`

```
{
  "projects": {
    "some-project": {
      "root": "",
      "projectType": "application",
      "architect": {
        "build": {},
        "serve": {},
        "extract-i18n": {},
        "test": {},
        "lint": {}
      }
    },
    "some-project-e2e": {}
  },
  "schematics": {
    "@schematics/angular:component": {
      "prefix": "app",
      "styleext": "scss"
    },
    "@schematics/angular:directive": {
      "prefix": "app"
    }
  }
}
```

```
{
  "projects": {
    "some-project": {
      "root": "",
      "projectType": "application",
      "architect": {
        "build": {},
        "serve": {},
        "extract-i18n": {},
        "test": {},
        "lint": {}
      }
    },
    "some-project-e2e": {}
  },
  "schematics": {
    "@schematics/angular:component": {
      "prefix": "app",
      "styleext": "scss"
    },
    "@schematics/angular:directive": {
      "prefix": "app"
    }
  }
}
```



## NEW – MULTIPLE APPS

6.0.x

- ▶ Can support multiple angular applications with one project
- ▶ Each app can have a "name"
  - ▶ Use `ng serve <name>`
- ▶ **Note** that `e2e`, `lint` and `test` are **per project!**

```
{
  "projects": {
    "some-project": {
      "root": "",
      "projectType": "application",
      "architect": {
        "build": {},
        "serve": {},
        "extract-i18n": {},
        "test": {},
        "lint": {}
      }
    },
    "some-project-e2e": {}
  },
  "schematics": {
    "@schematics/angular:component": {
      "prefix": "app",
      "styleext": "scss"
    },
    "@schematics/angular:directive": {
      "prefix": "app"
    }
  }
}
```

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

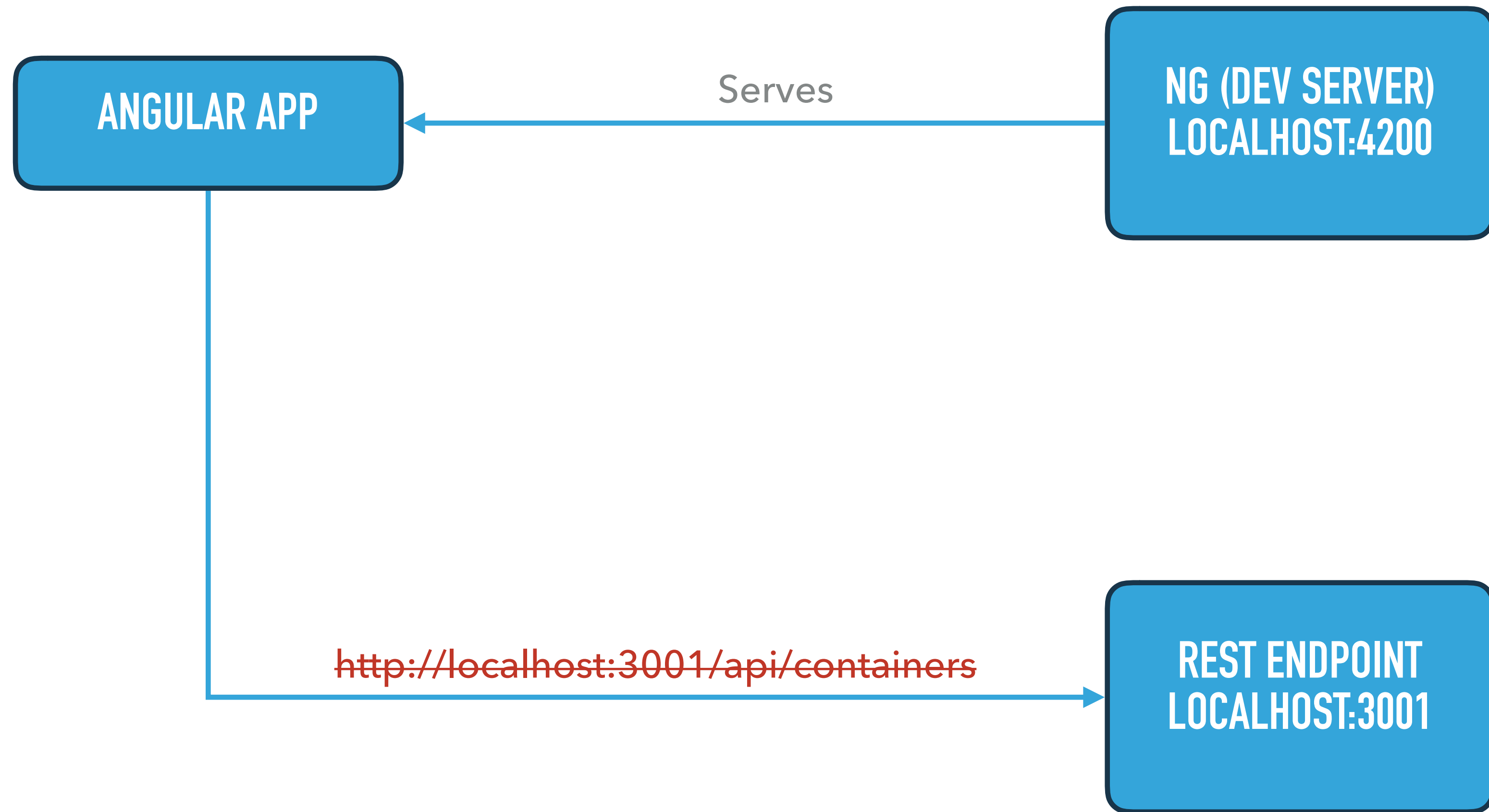
NG TEST

```
$ ng test --browser ChromeHeadless # watch mode
```

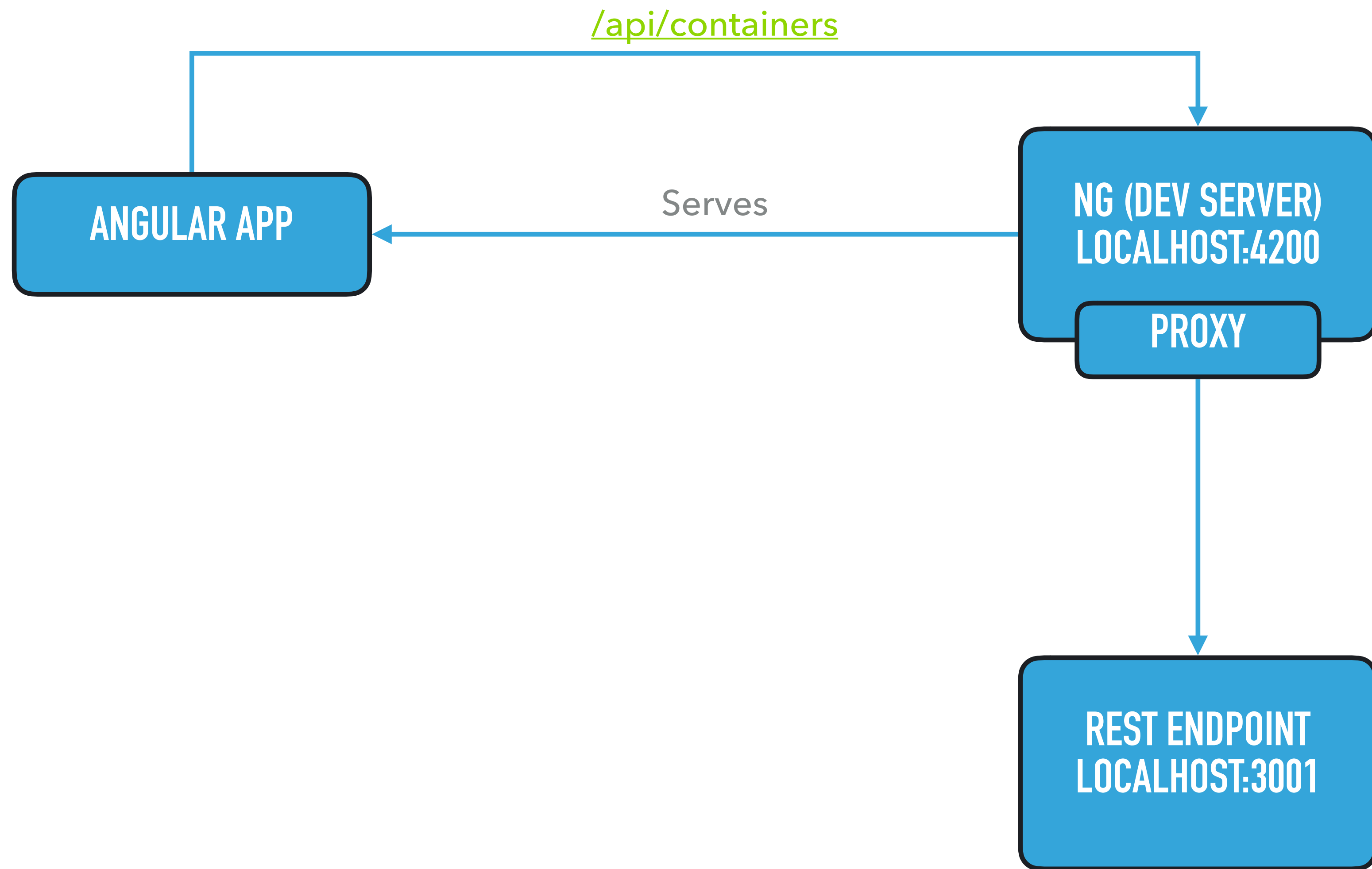
```
$ ng test --browser ChromeHeadless --single-run # ci
```

NG SERVE

CORS



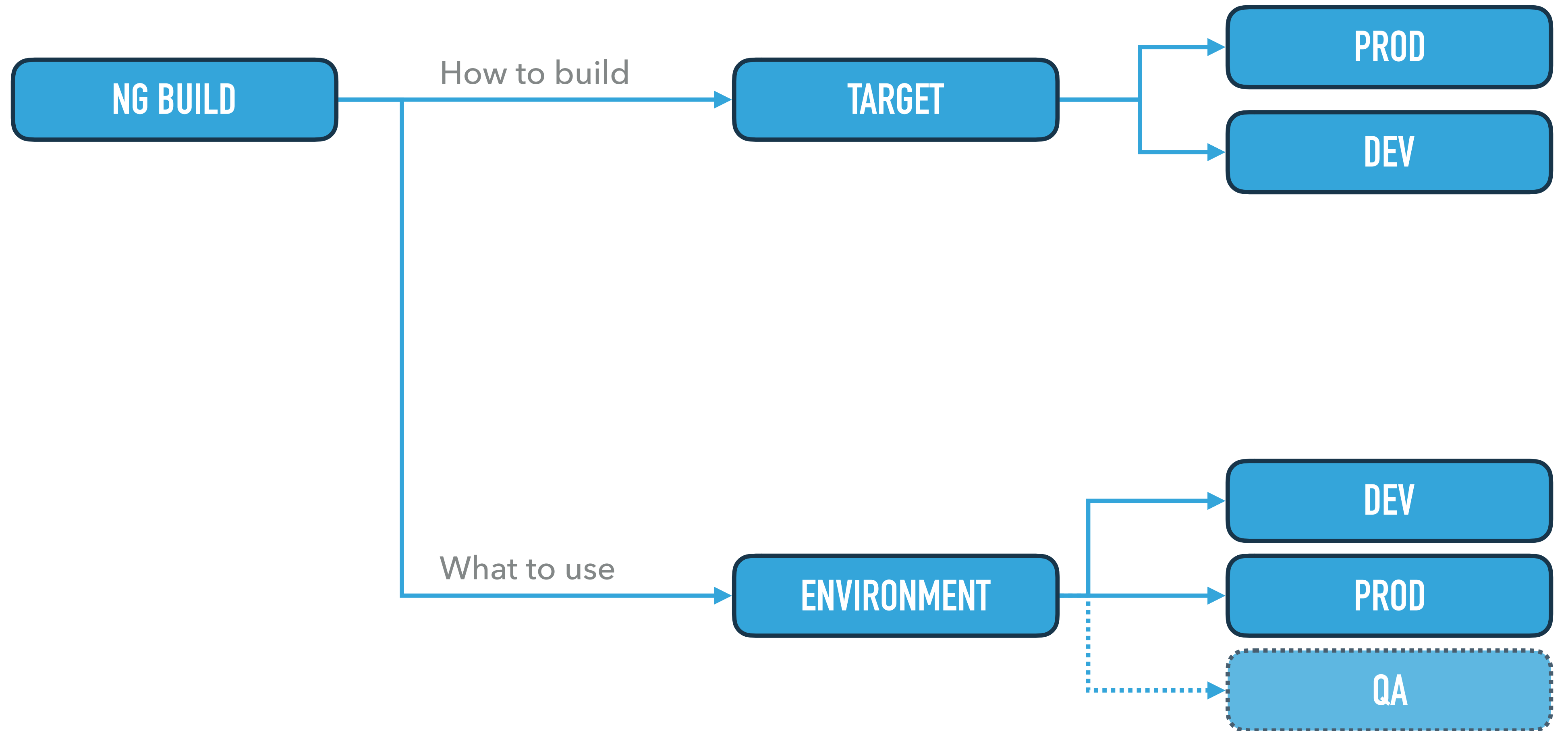




```
> cat proxy.config.json
{
  "/api": {
    "target": "http://localhost:3001",
    "secure": false
  }
}
> ng serve -pc proxy.config.json
```

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

NG BUILD



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

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

# BUILD

1.7.4

- ▶ `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`

```
{
  "projects": {
    "some-project": {
      "root": "",
      "projectType": "application",
      "architect": {
        "build": {
          "configurations": {
            "production": {
              "optimization": true,
              "sourceMap": false,
              "fileReplacements": [
                {
                  "src": "src/environments/environment.ts",
                  "replaceWith": "src/environments/environment.prod.ts"
                }
              ]
            }
          ]
        }
      }
    }
  }
}
```

# not sure what the difference here

ng build -c production;

ng build --prod



# 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**