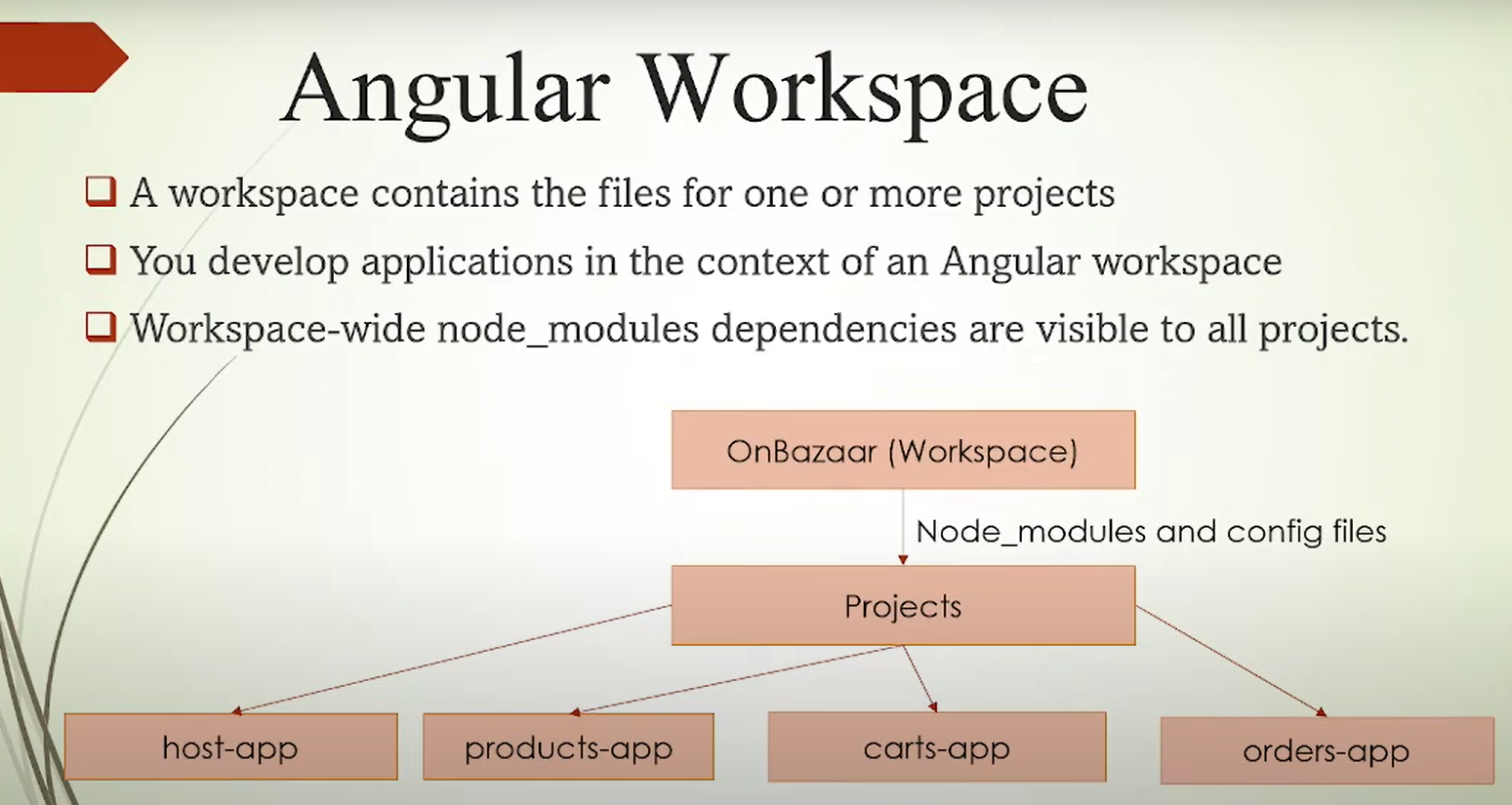
> cd C:\Dev\works\poc\microfront\module\_federal\sample1

## 1. Project Setup

### 1.1 Create Blank Angular workspace:

> ng new s1-root --create-application false



### 1.2 Create applications

> cd s1-root

> ng g application host-app

> ng g application products-app

> ng g application carts-app

> ng g application orders-app

## 2. Add Module Federation

### 2.1 Install bootstrap for styling

> npm install bootstrap

Go to <https://bootswatch.com/cerulean/> to get styles

### 2.2 Add Module Federation

> ng add @angular-architects/module-federation --project host-app --port 5000

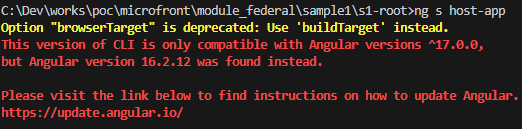
> ng add @angular-architects/module-federation --project carts-app --port 4000

> ng add @angular-architects/module-federation --project orders-app --port 3000

> ng add @angular-architects/module-federation --project products-app --port 2000

### 2.3 Start an app

> ng serve host-app



**Solution**:

Downgrade as below:

* @angular-architects/module-federation: From "^17.0.1" to "^16.0.1"
* ngx-build-plus: From "^17.0.0" to "^16.0.0"

### 2.4 Add some modules and pages for demo

> ng g m products --route products --module app.module --project products-app

> ng s products-app

Go to http://localhost:2000

## 3. Load first micro-frontend in host-app

### 3.2 Configure module federation

#### 3.1.1 Config webpack for products-app

./projects/products-app/Webpack.config.js

const { shareAll, withModuleFederationPlugin } = require('@angular-architects/module-federation/webpack');

module.exports = withModuleFederationPlugin({

  name: 'products',

  exposes: {

    './Module': './projects/products-app/src/app/products/products.module.ts',

  },

  shared: {

    ...shareAll({ singleton: true, strictVersion: true, requiredVersion: 'auto' }),

  }

});

#### 3.1.2 Config webpack for host-app

./projects/host-app/Webpack.config.js

const { shareAll, withModuleFederationPlugin } = require('@angular-architects/module-federation/webpack');

module.exports = withModuleFederationPlugin({

  remotes: {

    products: 'http://localhost:2000/remoteEntry.js',

  },

  shared: {

    ...shareAll({ singleton: true, strictVersion: true, requiredVersion: 'auto' }),

  }

});

#### 3.1.3 Display products app from host-app

./projects/host-app/src/types.d.ts

declare module 'products/Module';

./projects/host-app/src/app/app-routing.module.ts

const routes: Routes = [

  { path: 'products', loadChildren: () => import('products/Module').then(m => m.ProductsModule) }

];

./projects/host-app/src/app/app.component.html

  <a [routerLink]="['/products']" class="btn btn-primary">Products</a>

  <a [routerLink]="['/']" class="btn btn-primary">Home</a>

> ng s host-app

> ng s products-app

Go to <http://localhost:5000/products>

## 4. Load all Micro-Frontends in host-app

> ng g m orders --route orders --module app.module --project orders-app

> ng g m carts --route carts --module app.module --project carts-app

## 5. Add header layout to host-app

> ng g c layout --skip-tests --project host-app

## 6. List products

> ng g s services/core/network-request --skip-tests --project products-app

> ng g s services/functional/product --skip-tests --project products-app

https://fakestoreapi.com/