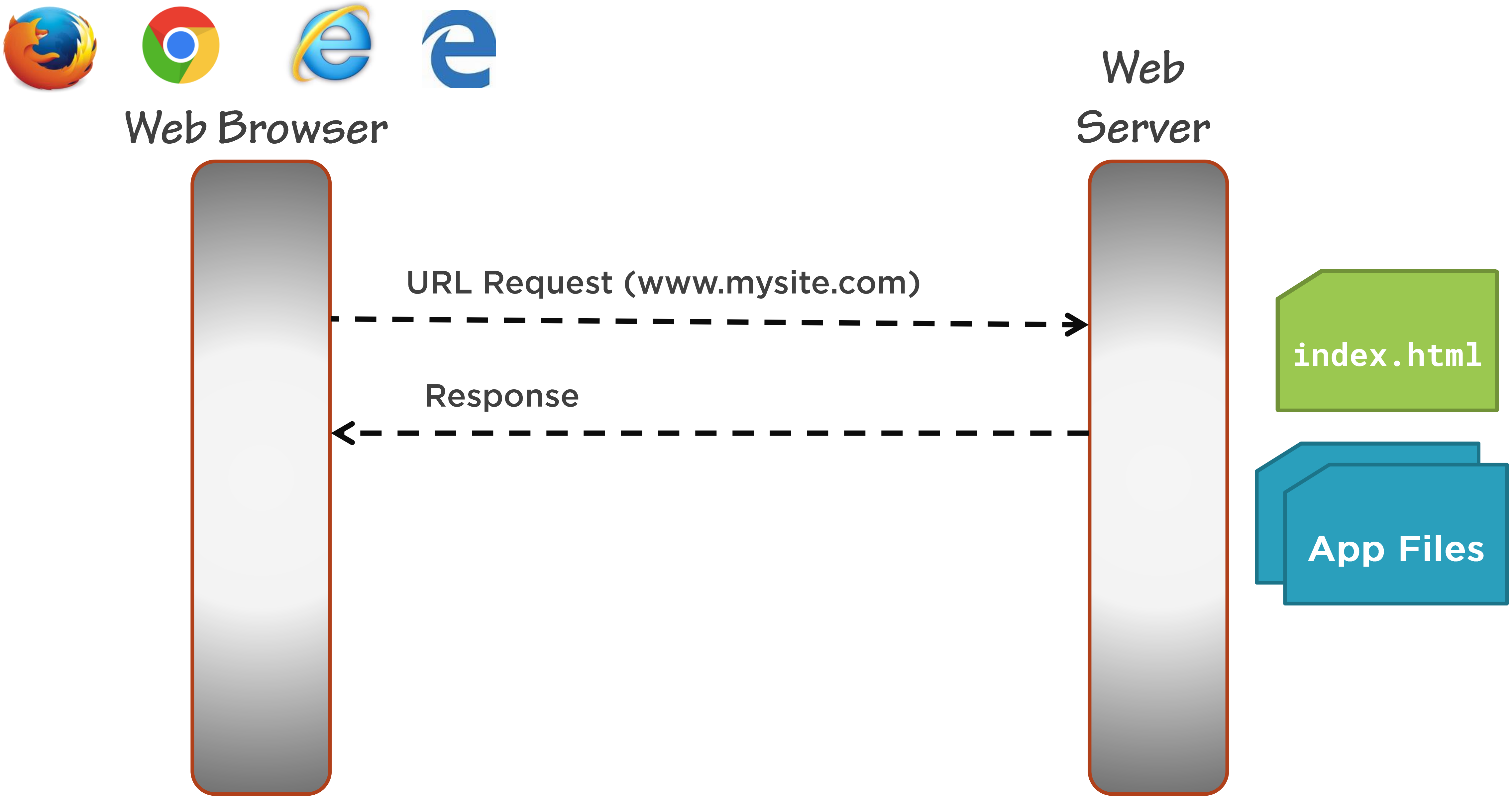
Lazy Loading

Deborah Kurata
CONSULTANT | SPEAKER | AUTHOR

@deborahkurata | blogs.msmvps.com/deborahk/



Web Browser

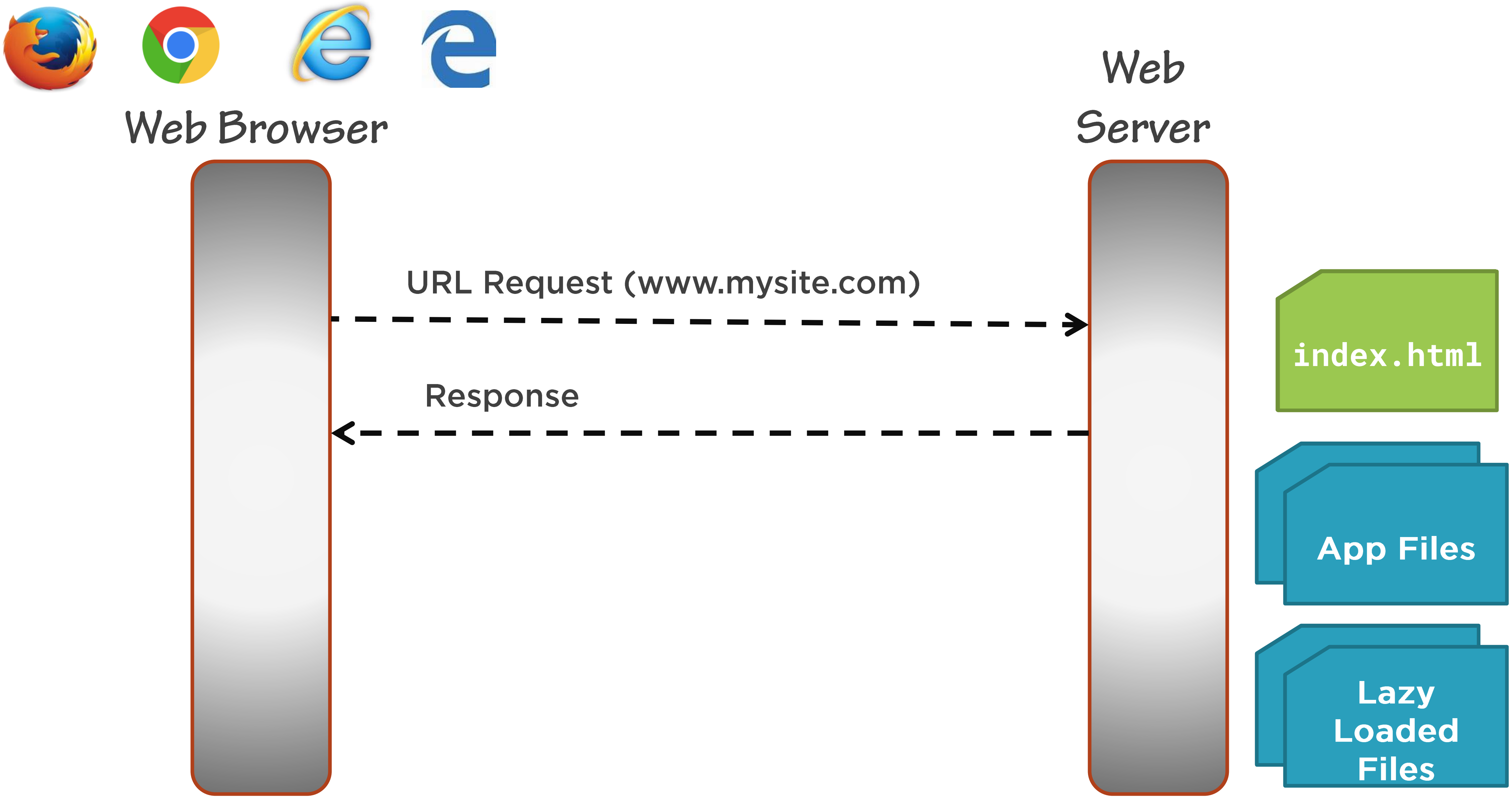
Web Server

URL Request (www.mysite.com)

Response

index.html

App Files



Web Browser

Web Server

URL Request (www.mysite.com)

Response

index.html

App Files

Lazy Loaded Files

Module Overview

Building and Serving Our Files

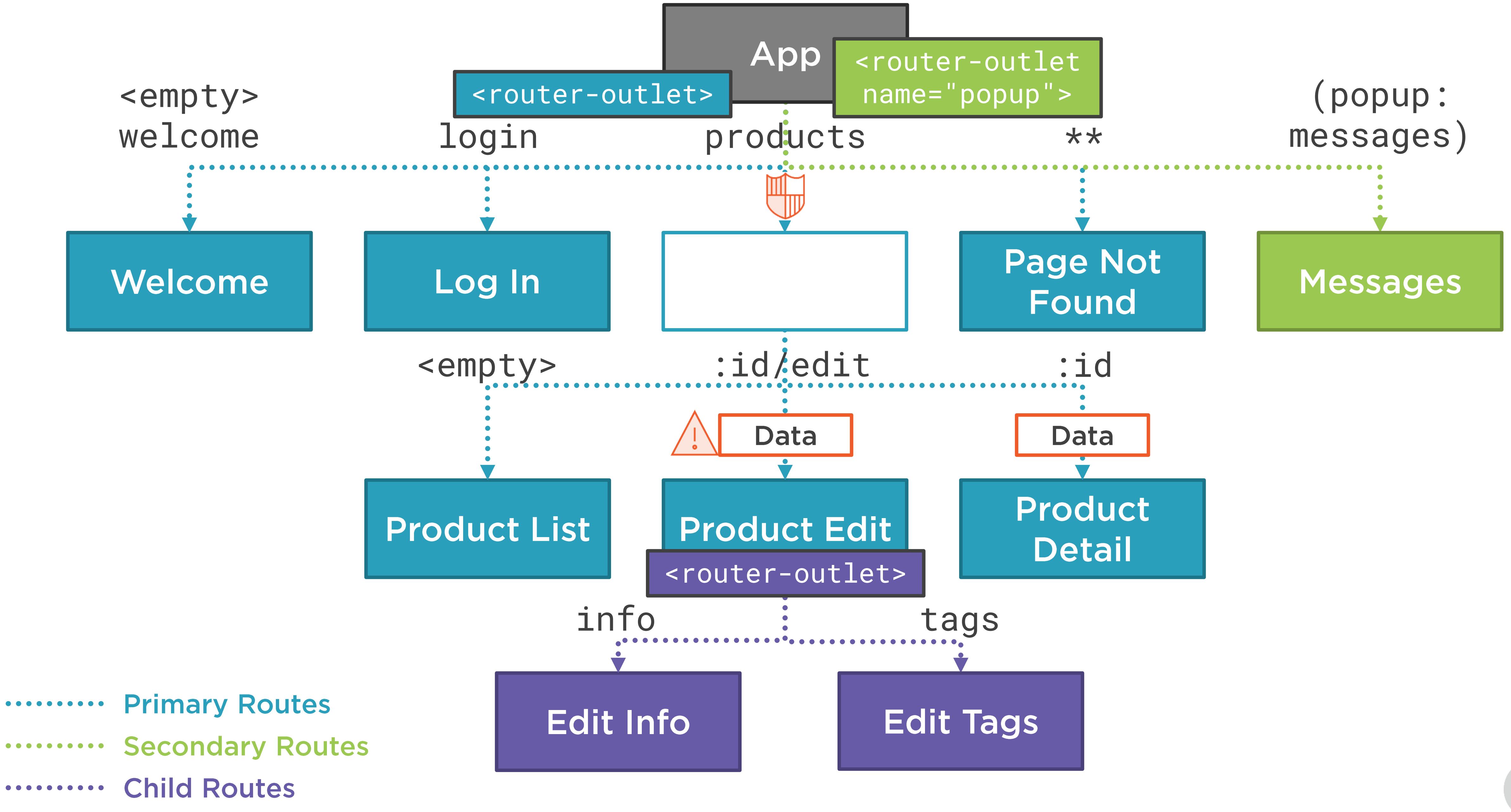
Preparing for Lazy Loading

Lazy Loading

canLoad Guard

Preloading Feature Modules

Custom Loading Strategies


App
<router-outlet>
<router-outlet name="popup">
<empty> welcome
login
products
**
(popup: messages)
Welcome
Log In
Page Not Found
Messages
<empty>
:id/edit
:id
Data
Data
Product List
Product Edit
<router-outlet>
Product Detail
info
tags
Edit Info
Edit Tags
Primary Routes
Secondary Routes
Child Routes

Demo

Building and Serving Our Files

Preparing for Lazy Loading

Use a feature module

Routes grouped under a single parent

Not imported in another module

Lazy Loading

app-routing.module.ts

```
RouterModule.forRoot([
 ...
 {
   path: 'products',
   loadChildren: () =>
      import('./products/product.module')
         .then(m => m.ProductModule)
 },
 ...
])
```

Lazy Loading

Delay downloading until needed

Speed up initial start time

canLoad Guard

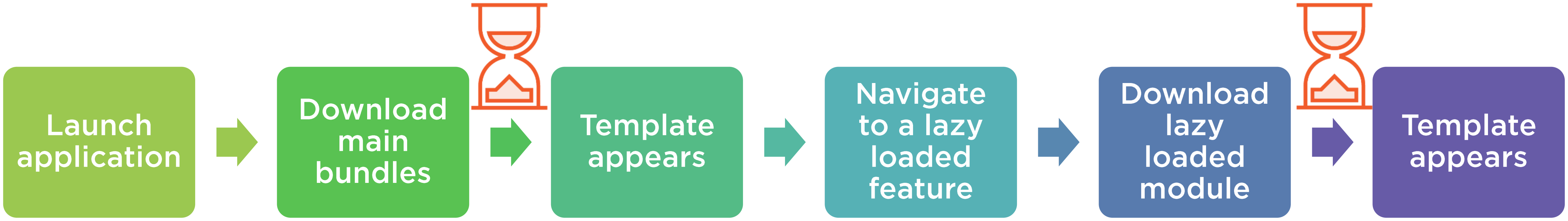
Checks criteria before loading an asynchronous route

Commonly used to:

- Prevent loading a route if a user cannot access it

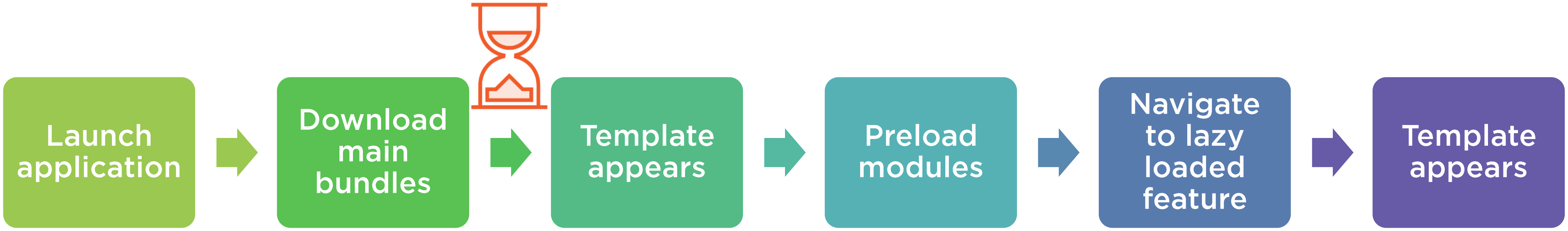
Preloading Feature Modules

Lazy Loading



Launch application → Download main bundles → Template appears → Navigate to a lazy loaded feature → Download lazy loaded module → Template appears

Preloading (Eager Lazy Loading)



Launch application → Download main bundles → Template appears → Preload modules → Navigate to lazy loaded feature → Template appears

Preload Strategies

No preloading
(on demand)

Preload all

Custom

Preload All Strategy

app-routing.module.ts

```
import { RouterModule, PreloadAllModules } from '@angular/router';
...

RouterModule.forRoot([
 { path: 'welcome', component: WelcomeComponent },
 {
   path: 'products',
   loadChildren: () =>
      import('./products/product.module').then(m => m.ProductModule)
 },
 ...
], { preloadingStrategy: PreloadAllModules })
```

Demo

Preloading Feature Modules

canLoad guard blocks preloading

Preload Strategies

No preloading
(on demand)

Preload all

Custom

Custom Preloading Strategy

1. Build a preloading strategy service

2. Set the preloading strategy routing option

Building a Preload Strategy Service

selective-strategy.service.ts

```
import { Injectable } from '@angular/core';
import { PreloadingStrategy } from '@angular/router';
import { Observable } from '@angular/router';

@Injectable({
  providedIn: 'root'
})
export class SelectiveStrategy implements PreloadingStrategy {

    preload(route: Route, load: Function): Observable<any> {
      ...
    }
}
```

Enabling a Strategy

app-routing.module.ts

```
...
import { SelectiveStrategy } from './selective-strategy.service';

@NgModule({
  imports: [
    RouterModule.forRoot([
      { path: 'welcome', component: WelcomeComponent },
      {
         path: 'products',
         loadChildren: () =>
            import('./products/product.module').then(m => m.ProductModule)
       }, ...
    ], { preloadingStrategy: SelectiveStrategy }),
  ]
})
export class AppModule { }
```

Lazy Loading Checklist: Preparing

Use a feature module

Group routes under a single parent

Don't import feature module

Lazy Loading Checklist: Configuring

Add the `loadChildren` property

```
{
  path: 'products',
  loadChildren: () =>
     import('./products/product.module')
         .then(m => m.ProductModule)
}
```

Lazy Loading Checklist: Preloading

Preload all modules with `PreloadAllModules`

Or build a custom preloading strategy service

```
RouterModule.forRoot([
 {
   path: 'products',
   loadChildren: () =>
      import('./products/product.module')
          .then(m => m.ProductModule)
 }, ...
], { preloadingStrategy: PreloadAllModules
})
```

Summary

Building and Serving Our Files

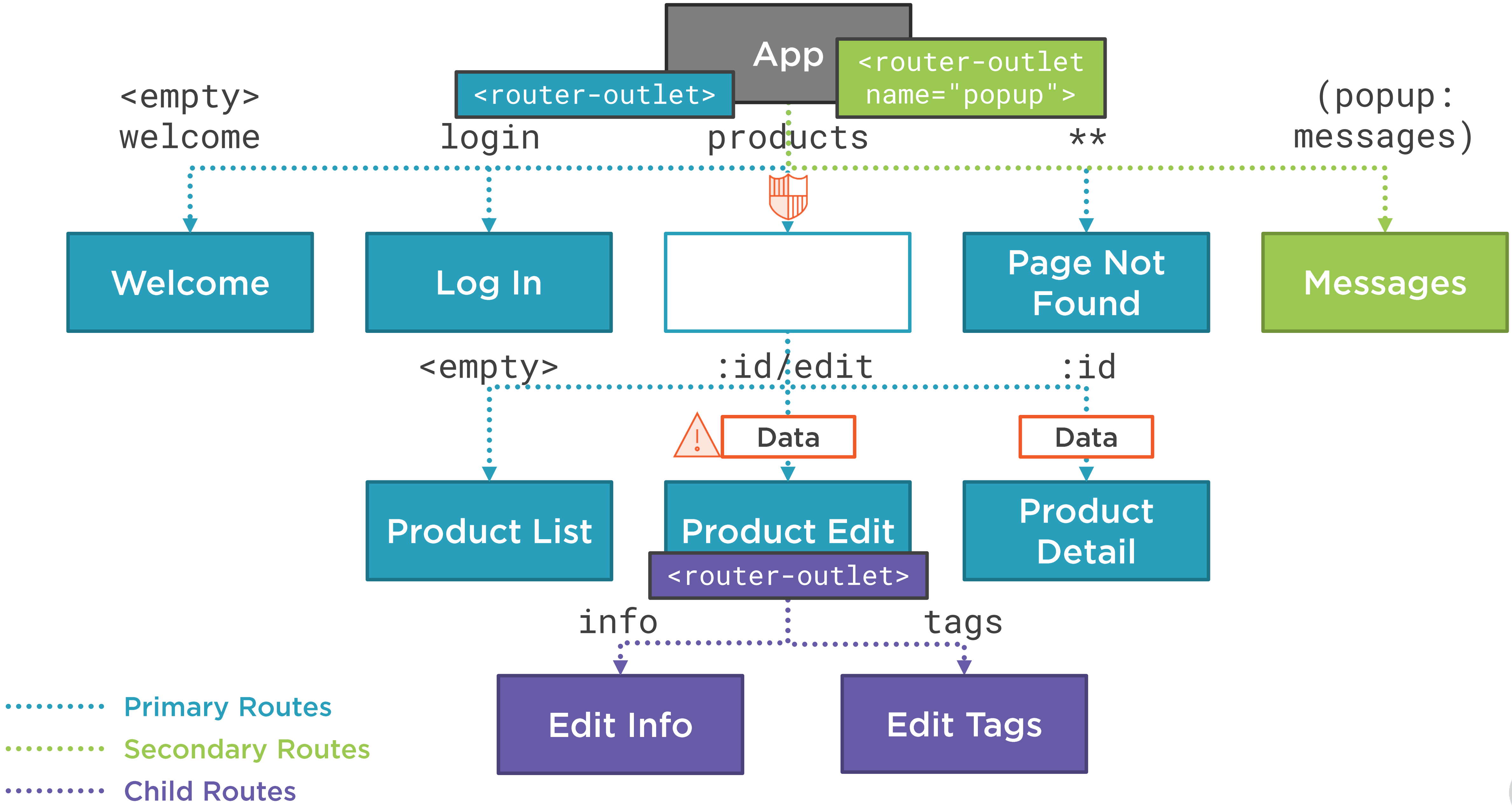
Preparing for Lazy Loading

Lazy Loading

canLoad Guard

Preloading Feature Modules

Custom Loading Strategies


App
<router-outlet>
<router-outlet name="popup">
<empty> welcome
login
products
**
(popup: messages)
Welcome
Log In
Page Not Found
Messages
<empty>
:id/edit
:id
Data
Data
Product List
Product Edit
<router-outlet>
Product Detail
info
tags
Edit Info
Edit Tags
Primary Routes
Secondary Routes
Child Routes