App Frameworks

#WWDC16

Optimizing On-Demand Resources

Session 221

Bill Bumgarner tvOS Engineering

Agenda

Agenda

Overview

Assigning Tags

Using BundleResourceRequest

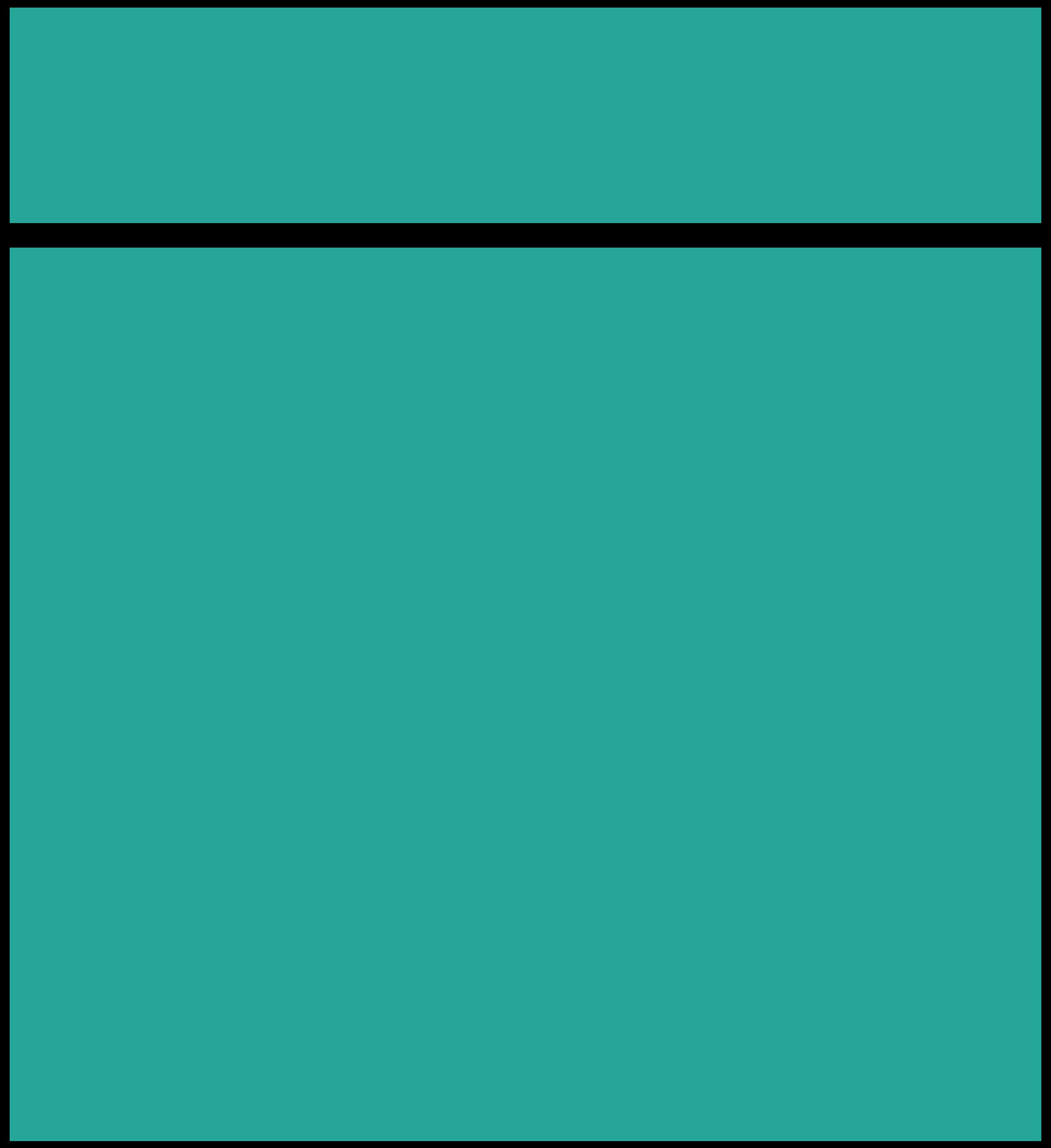
Optimizing First Launch

Optimizing Predictive Loading

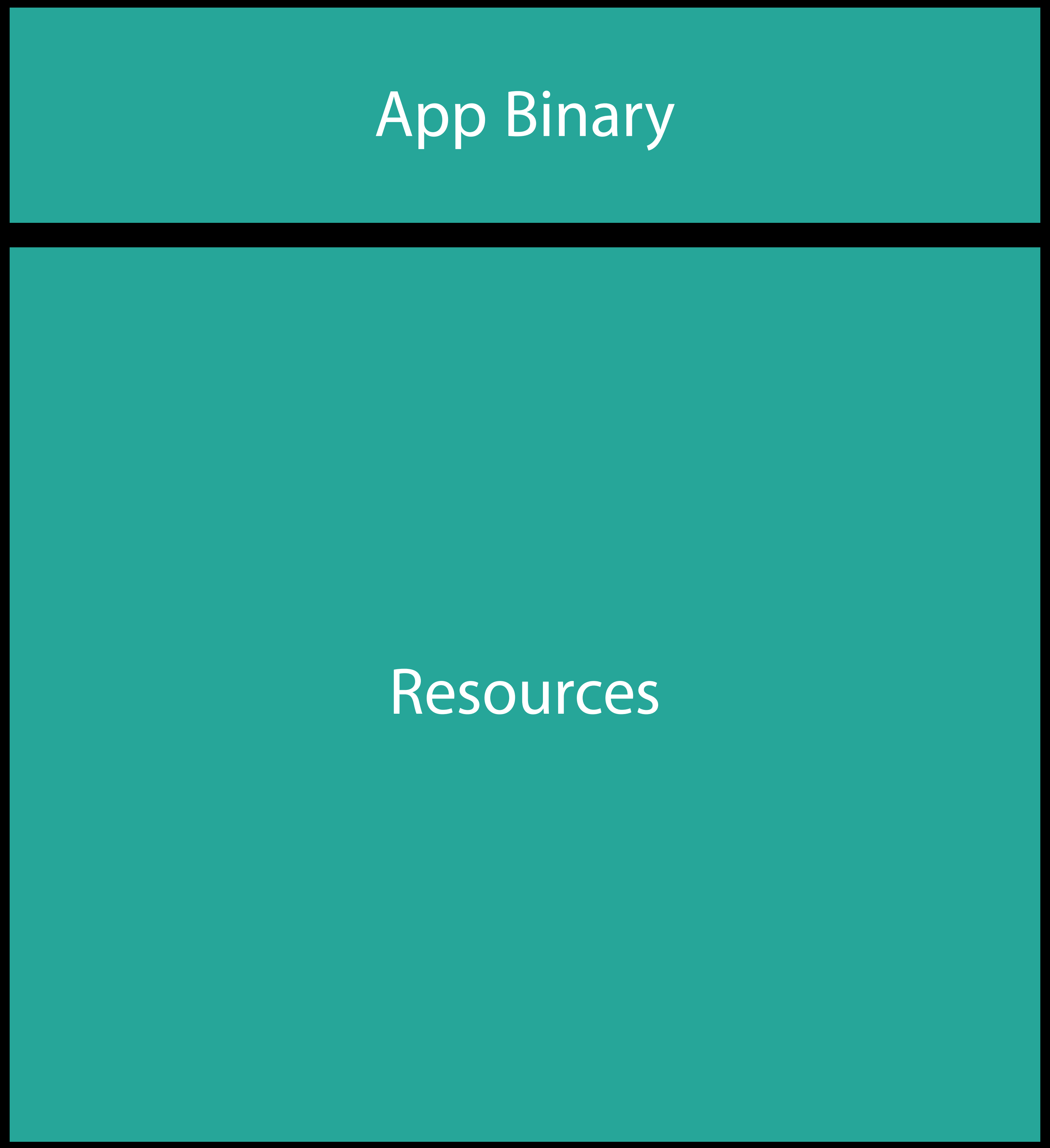
Optimizing App Updates

Motivation

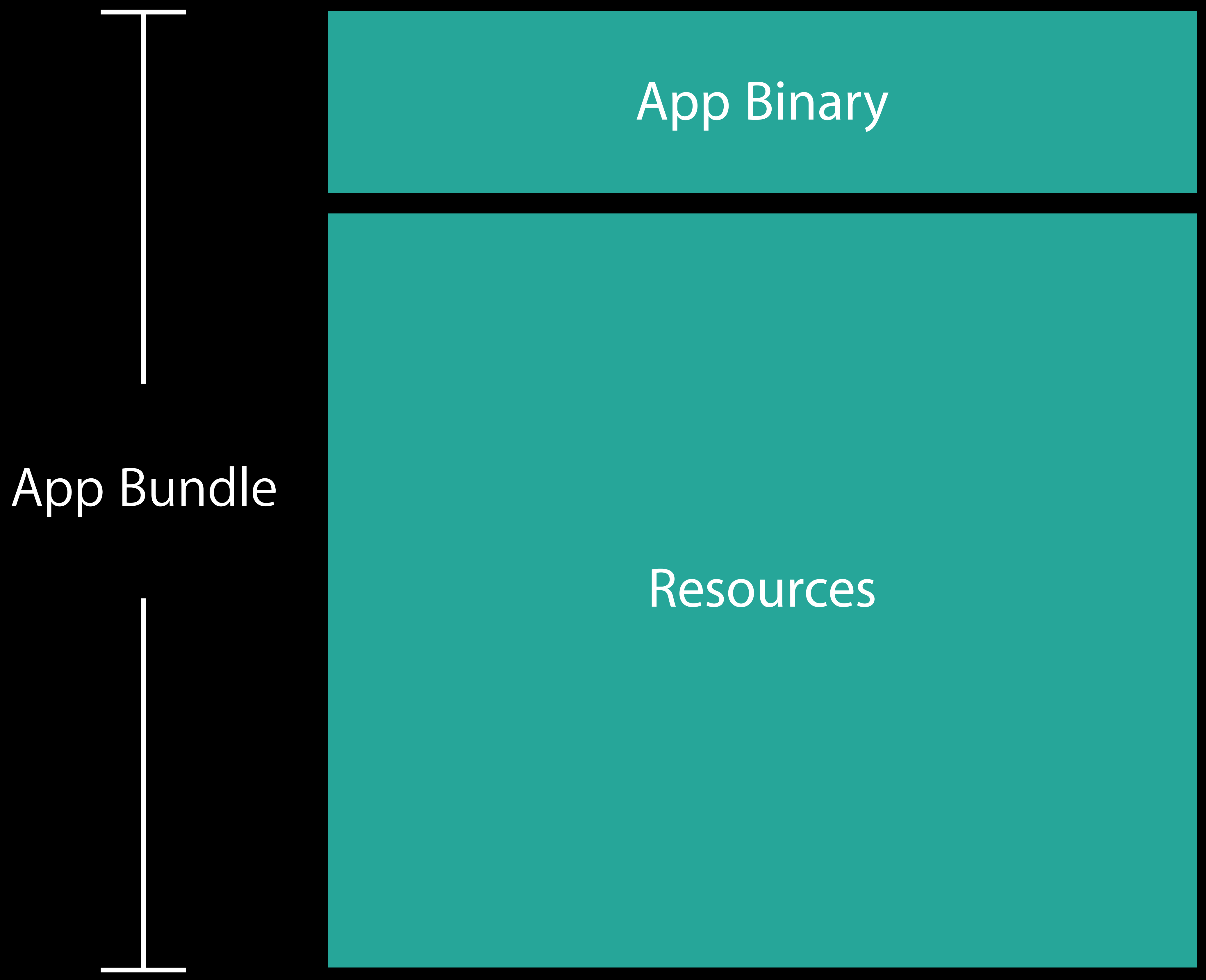
Traditional App



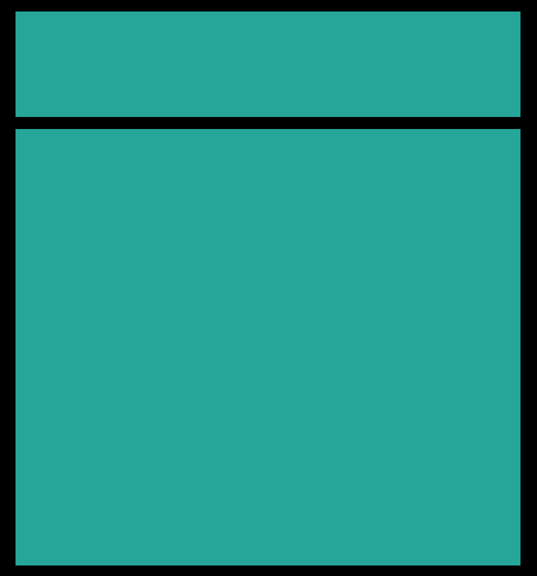
Traditional App



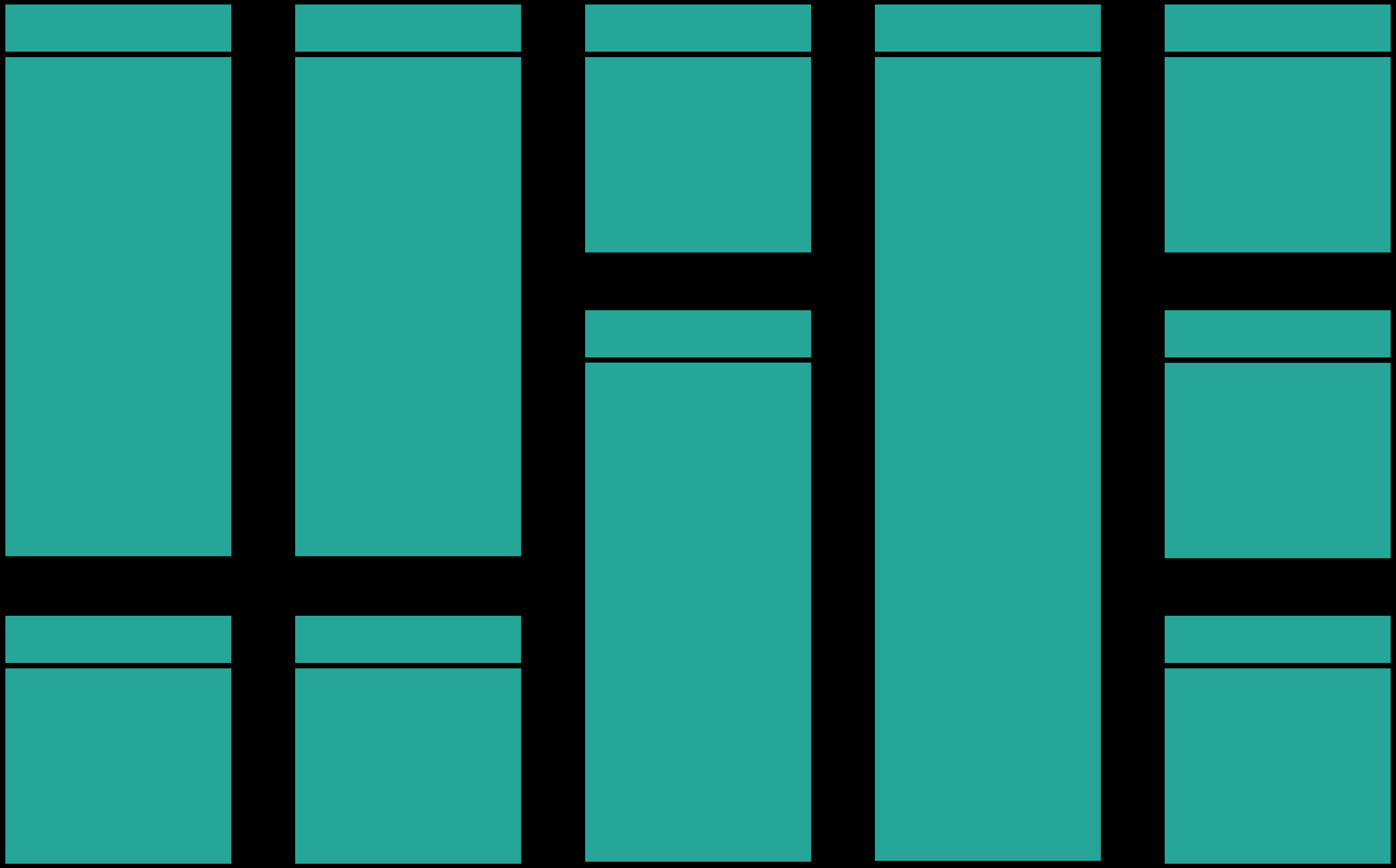
Traditional App



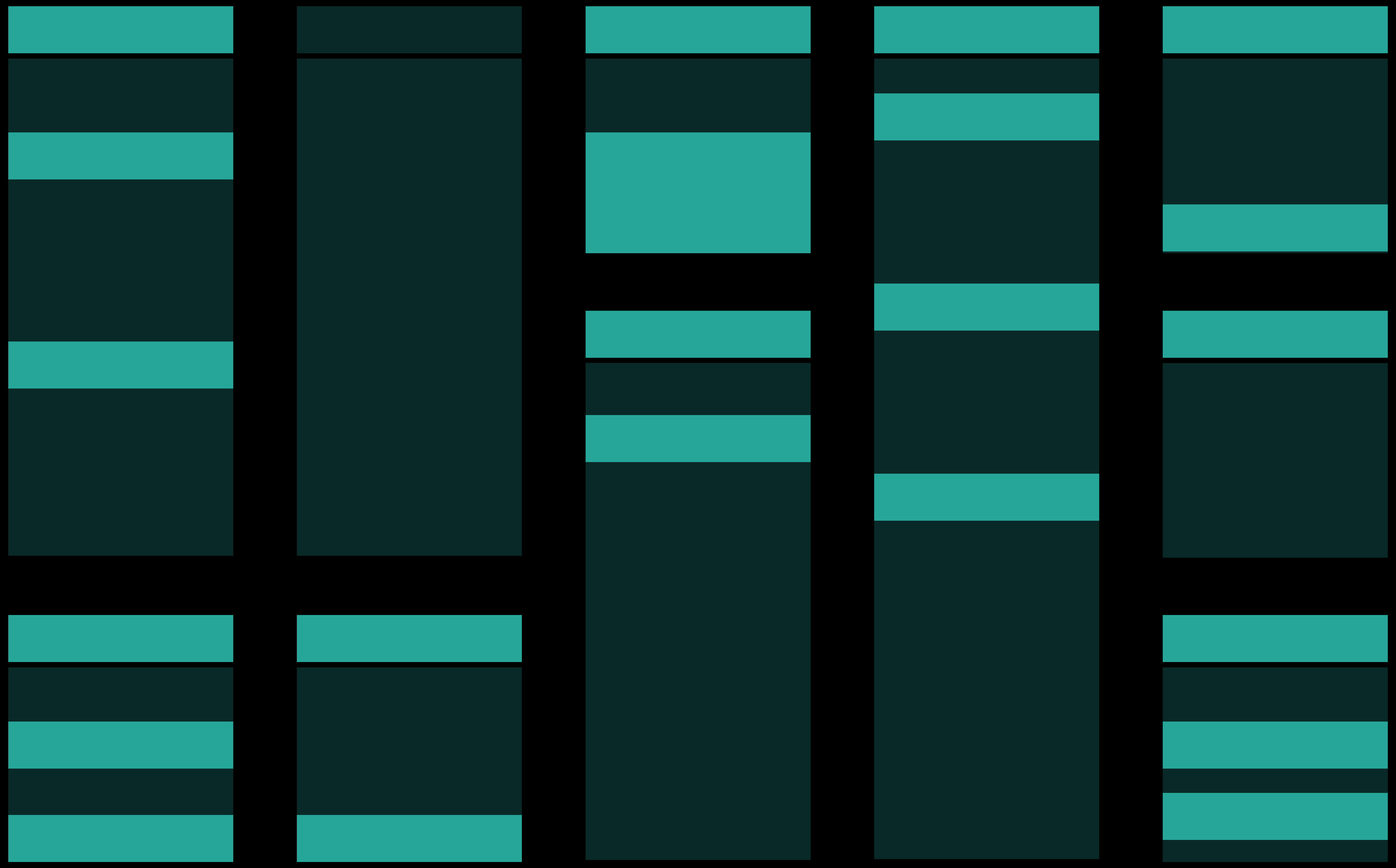
Traditional App



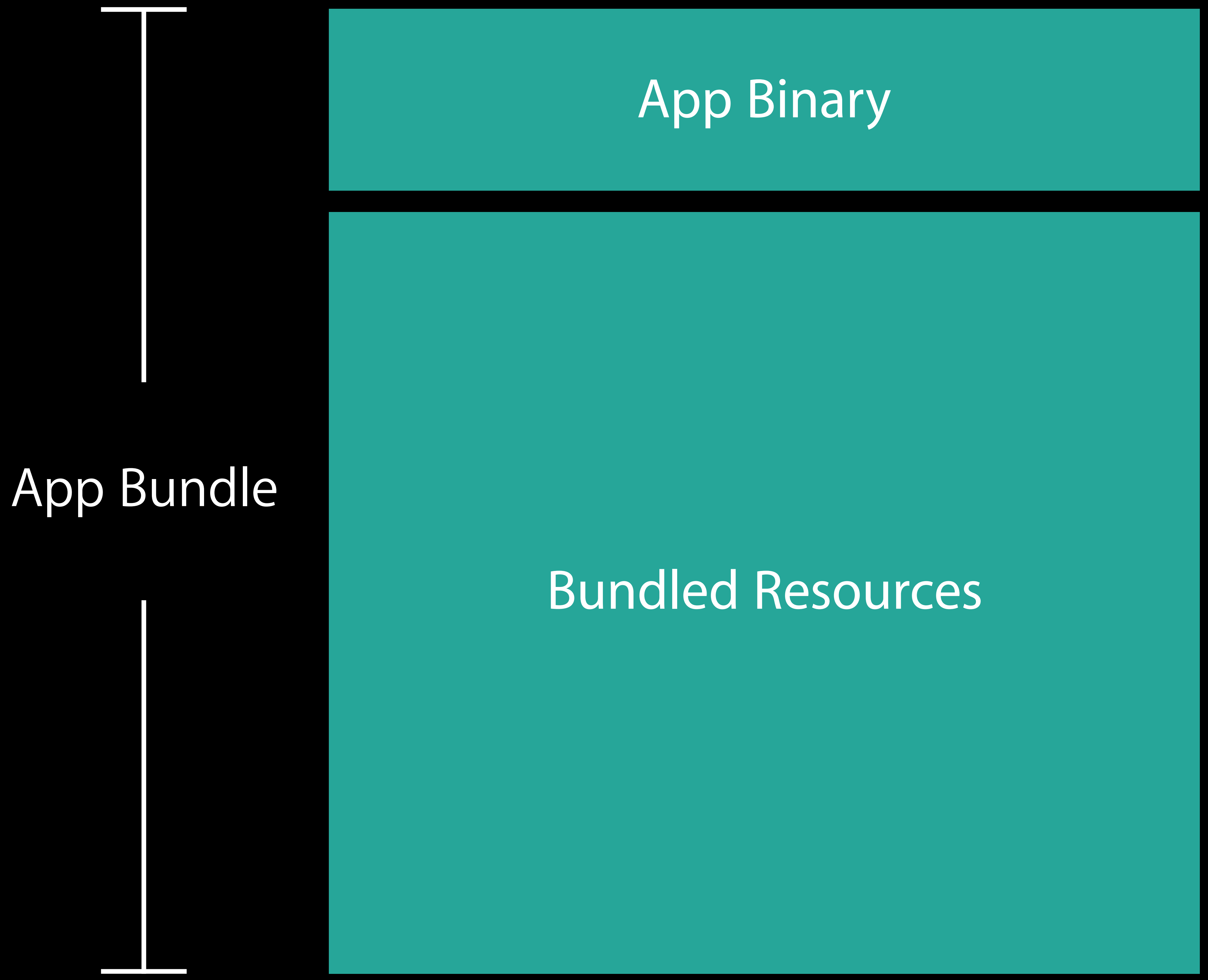
Traditional App



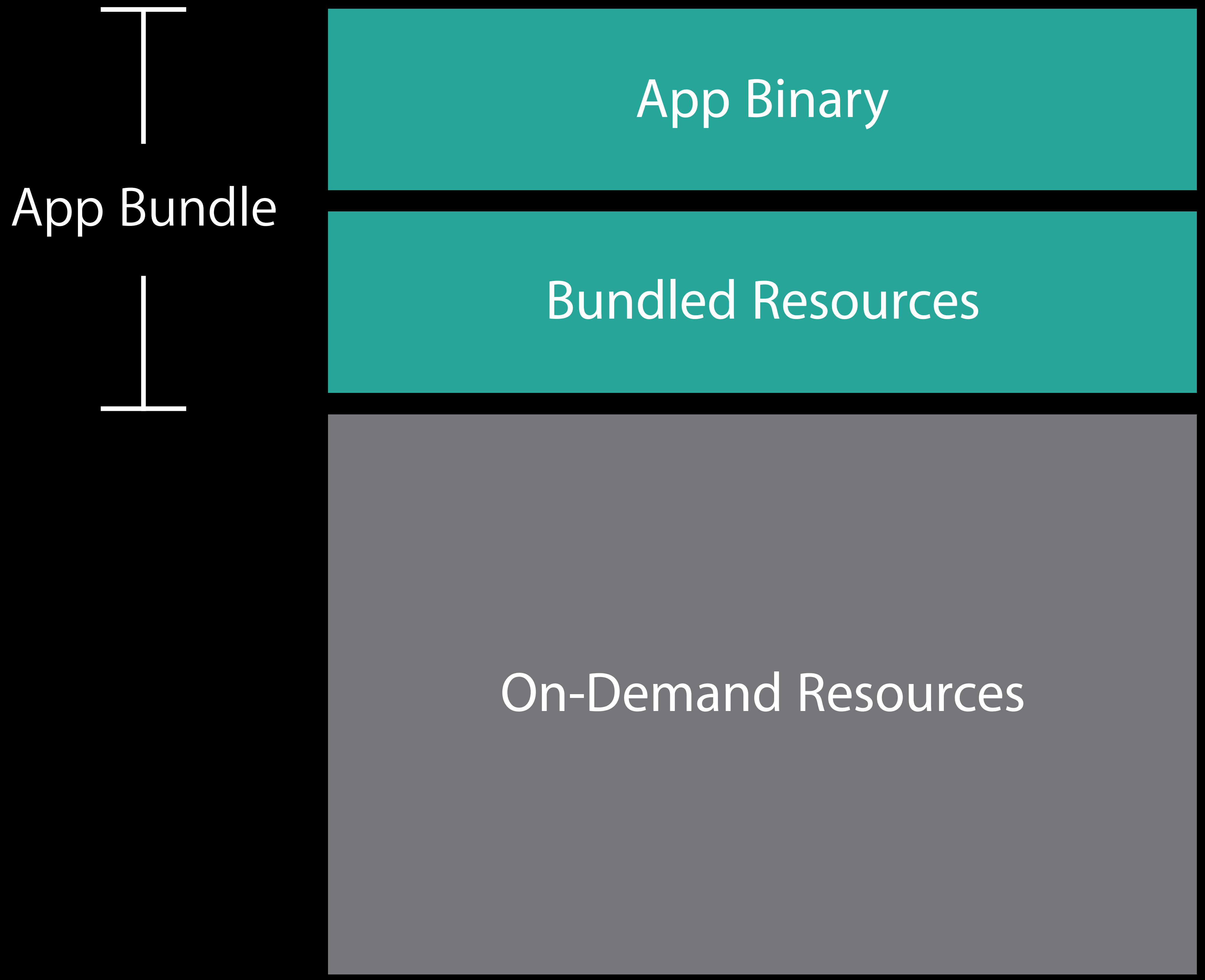
Traditional App



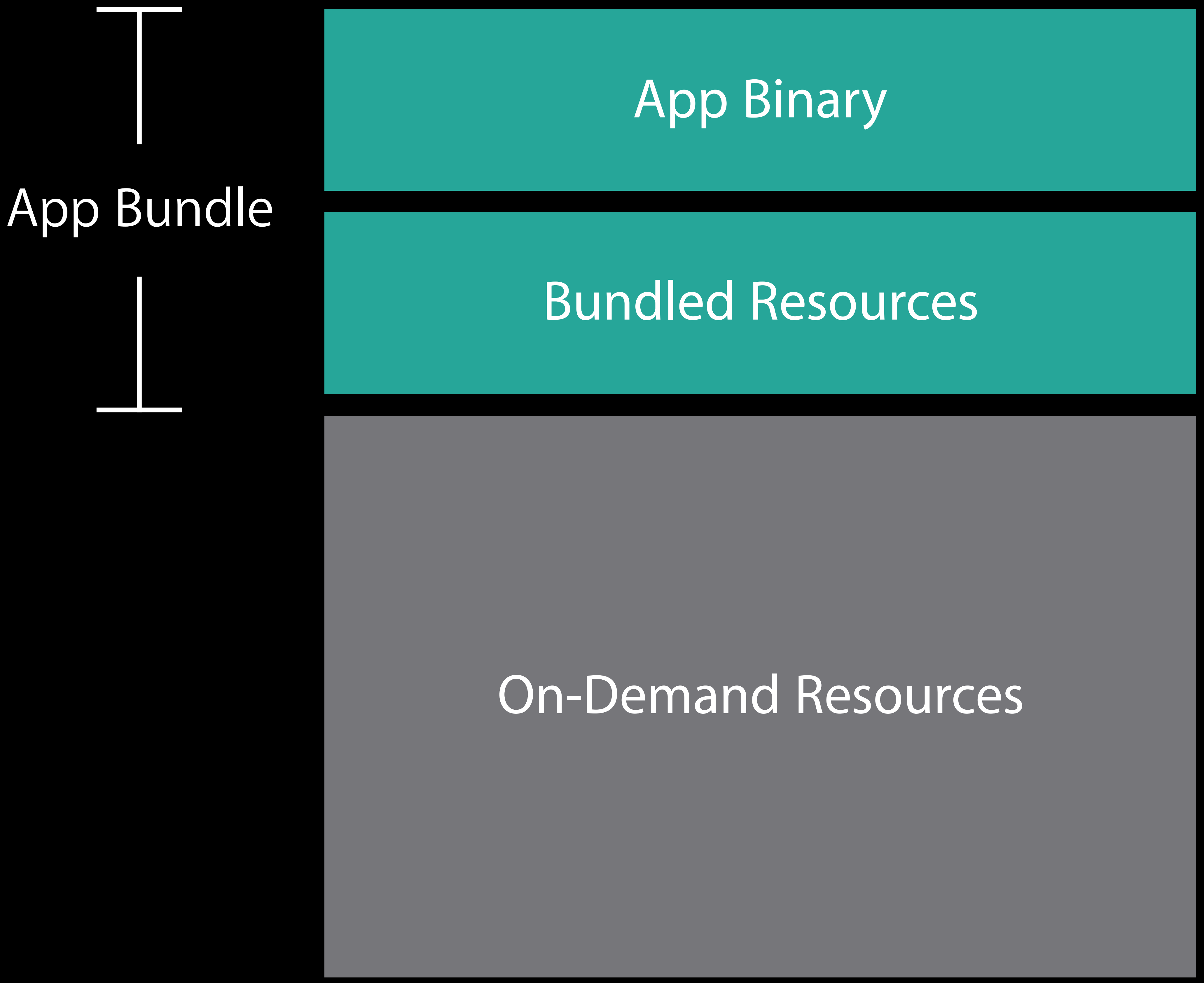
On-Demand Resources App



On-Demand Resources App



On-Demand Resources App



tvOS	iOS
Up to 200MB	Up to 4GB
Up to 20GB	Up to 20GB

On-Demand Resources

Dynamically loaded content

On-Demand Resources

Dynamically loaded content

Hosted on the App Store

On-Demand Resources

Dynamically loaded content

Hosted on the App Store

Downloadable during app install and by request

On-Demand Resources

- Dynamically loaded content
- Hosted on the App Store
- Downloadable during app install and by request
- Prioritized downloads

On-Demand Resources

Dynamically loaded content

Hosted on the App Store

Downloadable during app install and by request

Prioritized downloads

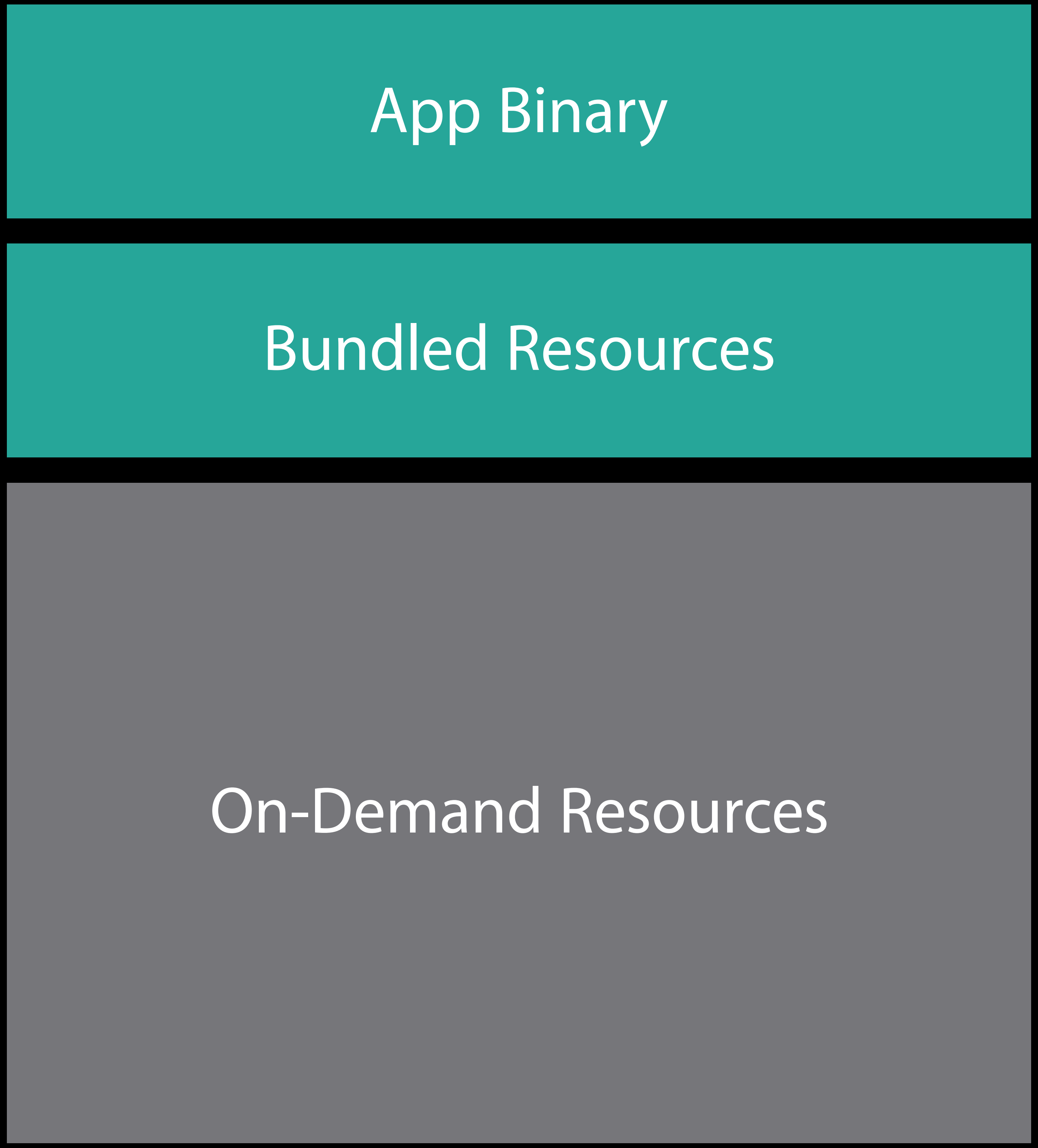
Intelligent content caching

On-Demand Resources

Benefits

Smaller main app bundle

- Faster initial download



On-Demand Resources

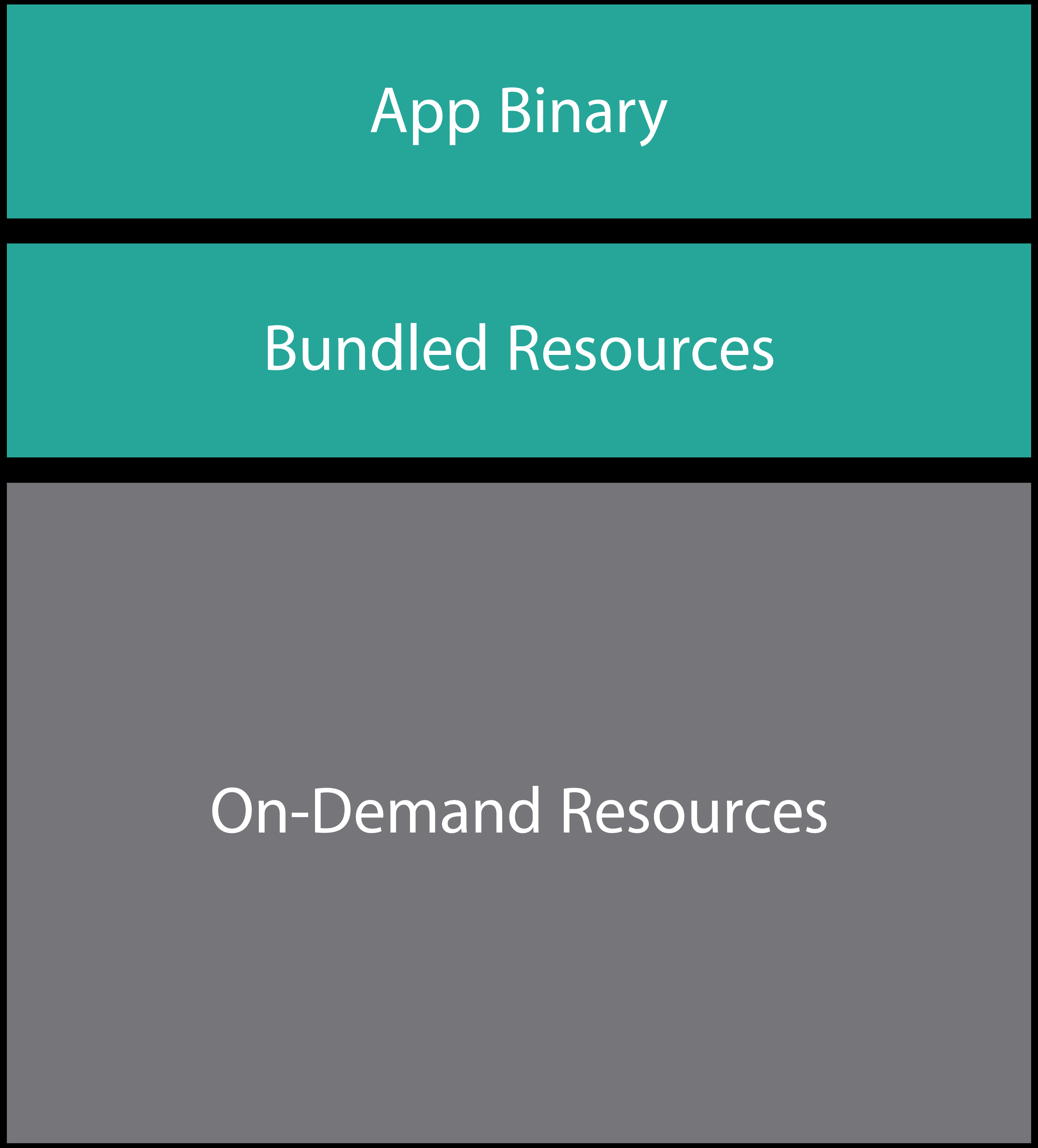
Benefits

Smaller main app bundle

- Faster initial download

Richer app content

- Up to 20-GB available on demand



On-Demand Resources

Benefits

Smaller main app bundle

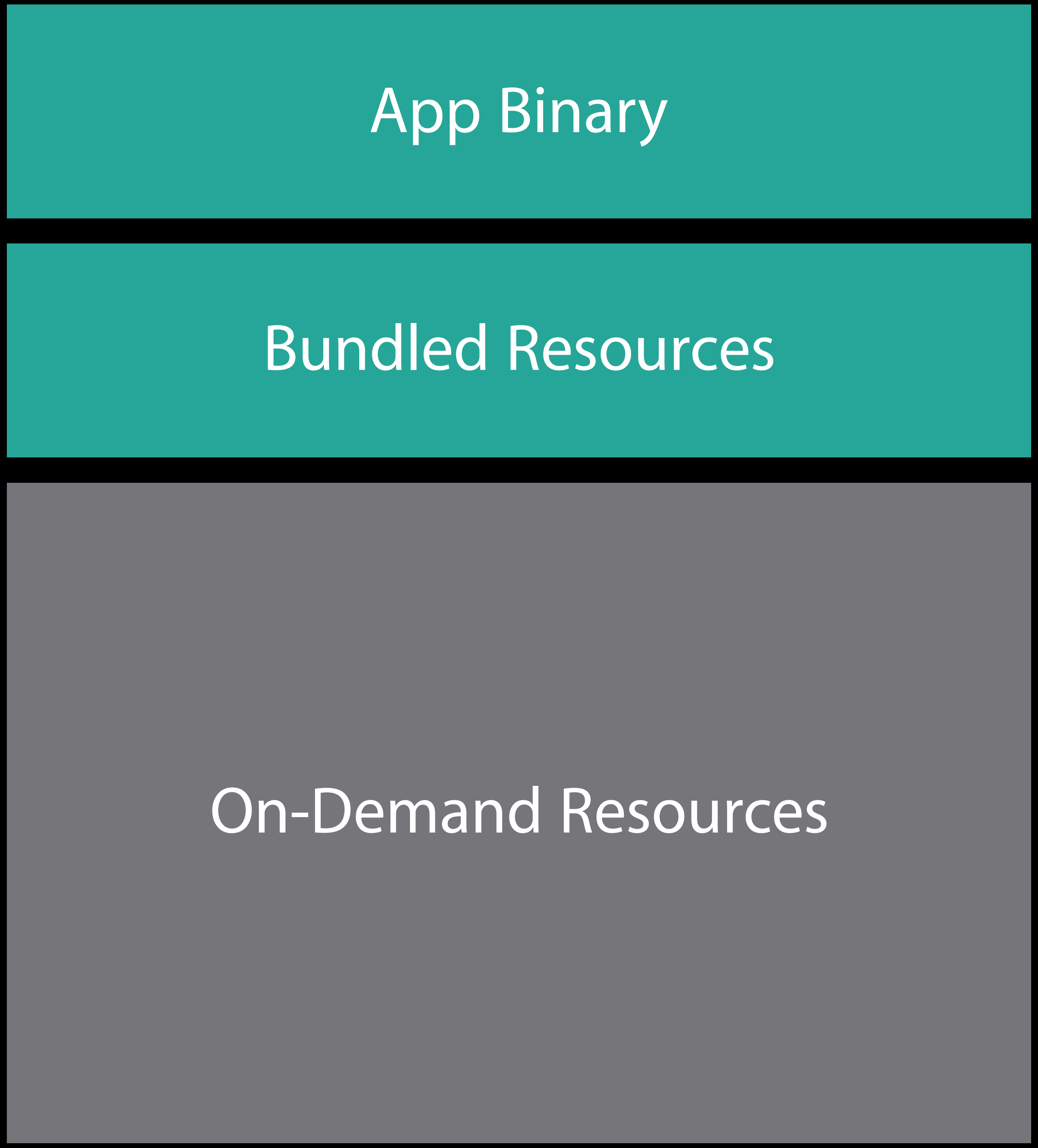
- Faster initial download

Richer app content

- Up to 20-GB available on demand

More apps installed and ready to run

- Reduces need to manage storage

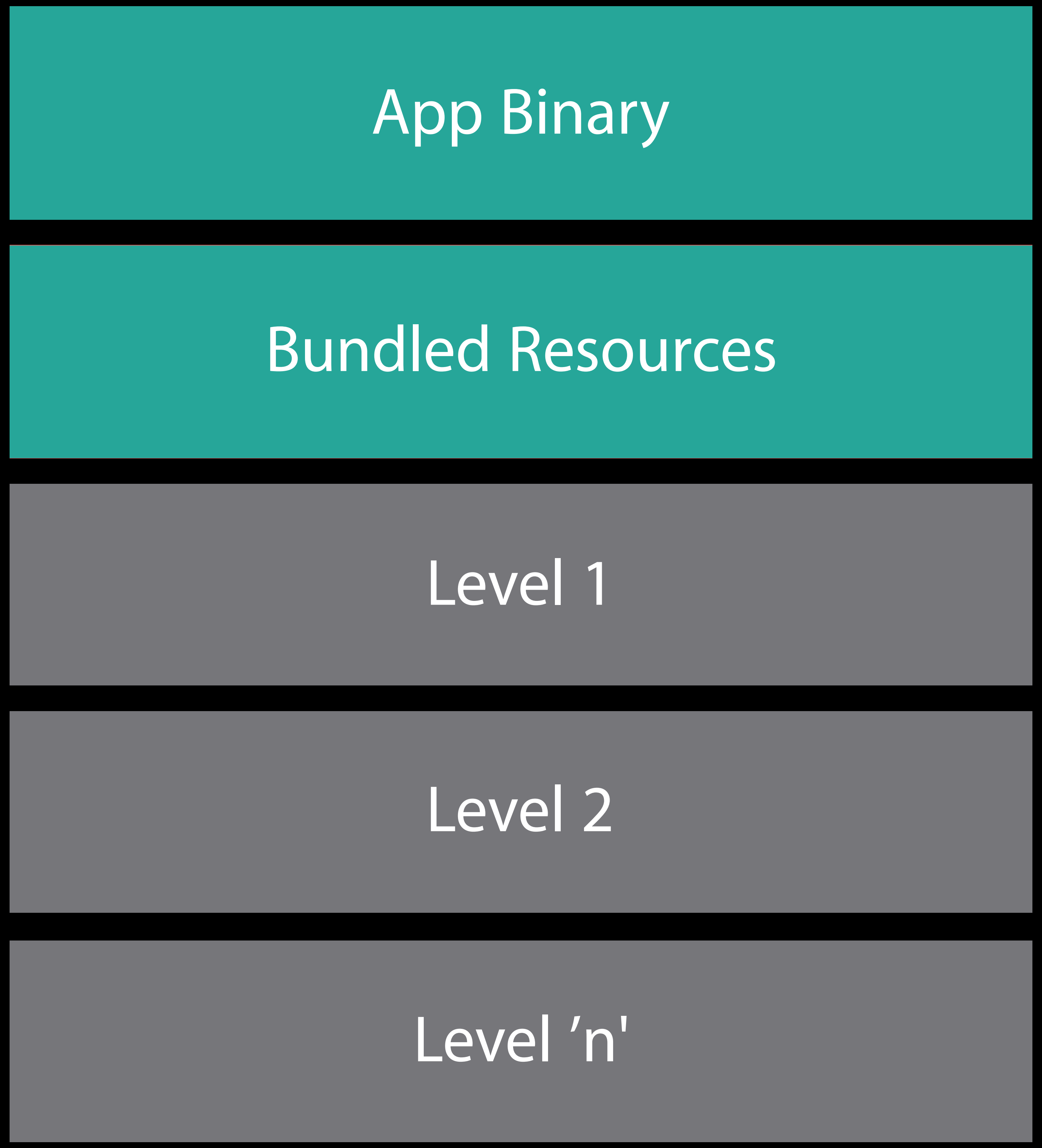


Assigning Tags

Assigning Tags
Packaging for download

Organize your resources

- By role within app
- By when you need them



Assigning Tags

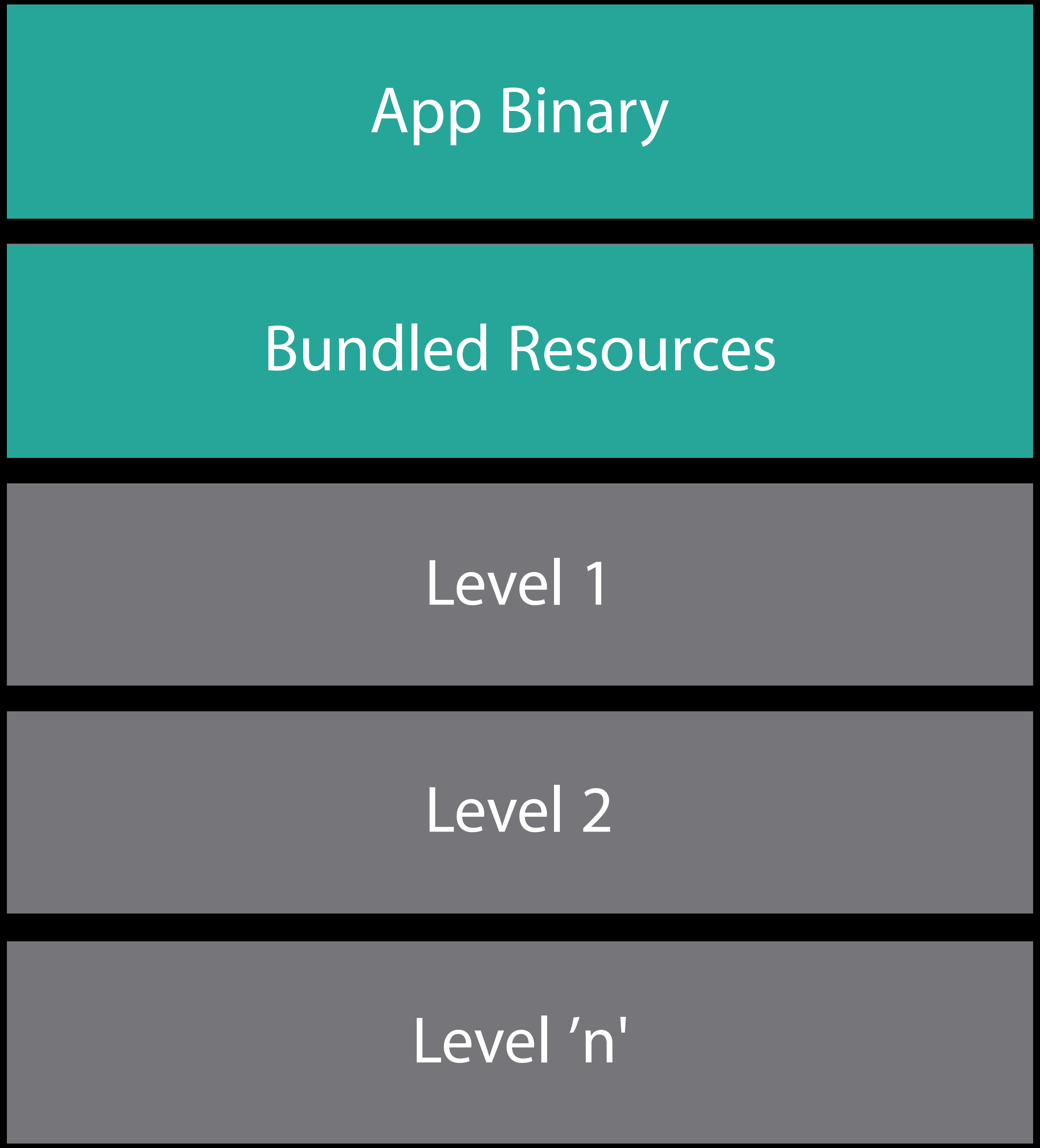
Packaging for download

Organize your resources

- By role within app
- By when you need them

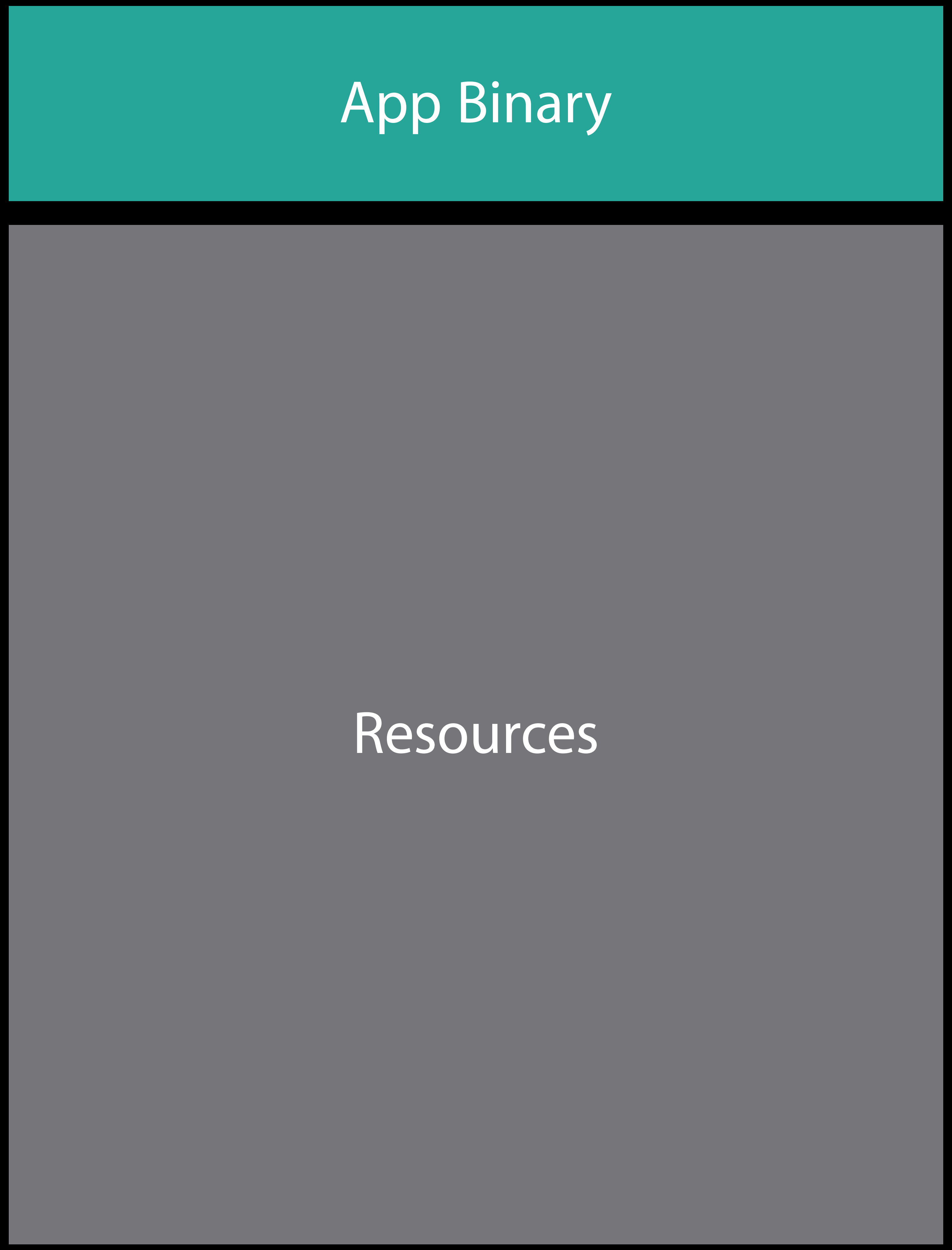
Apply tags using Xcode

- Like "Level 1"
- Tags are simple strings
- May apply to single assets or entire folders



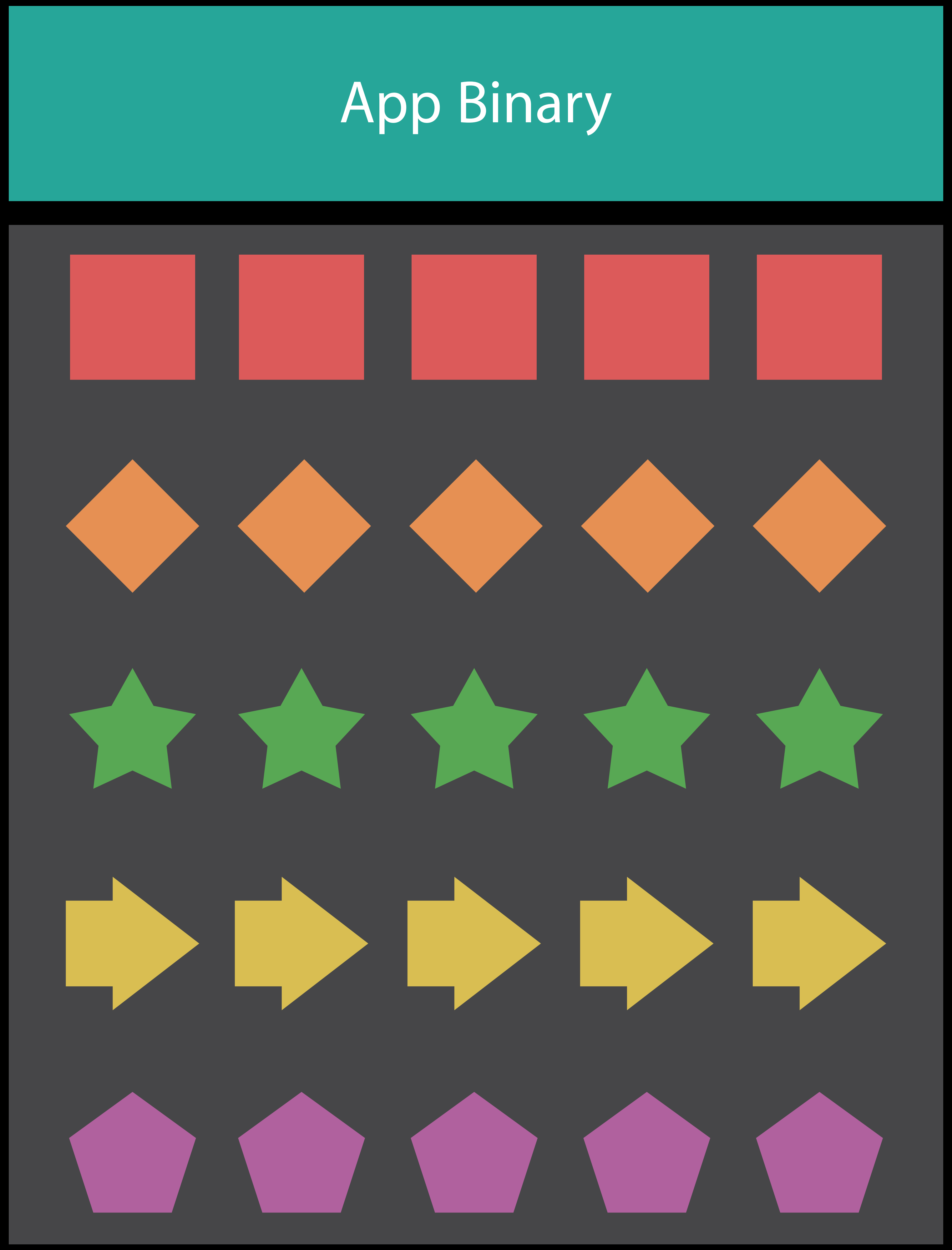
Resources

GreatGame.app



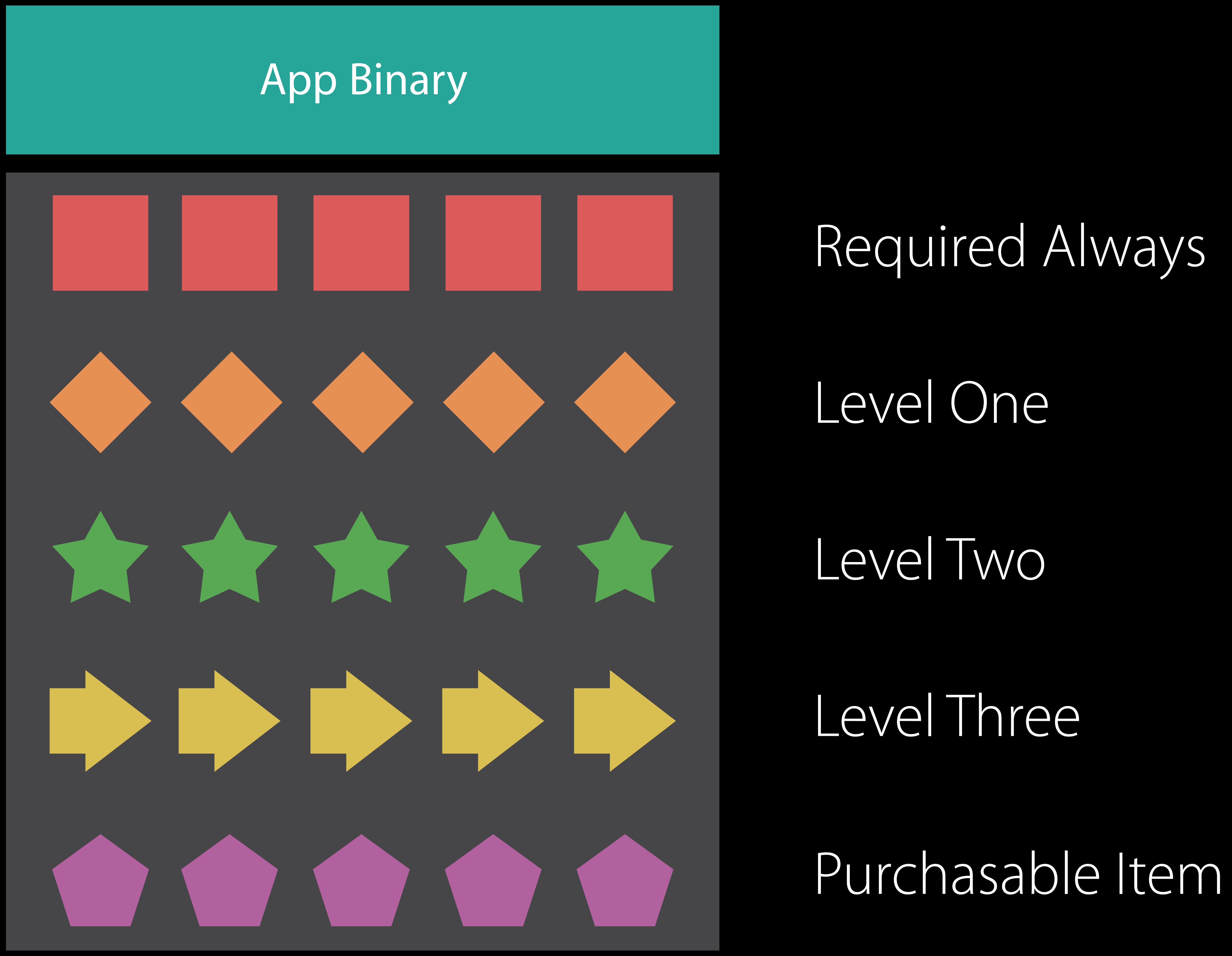
Resources

GreatGame.app



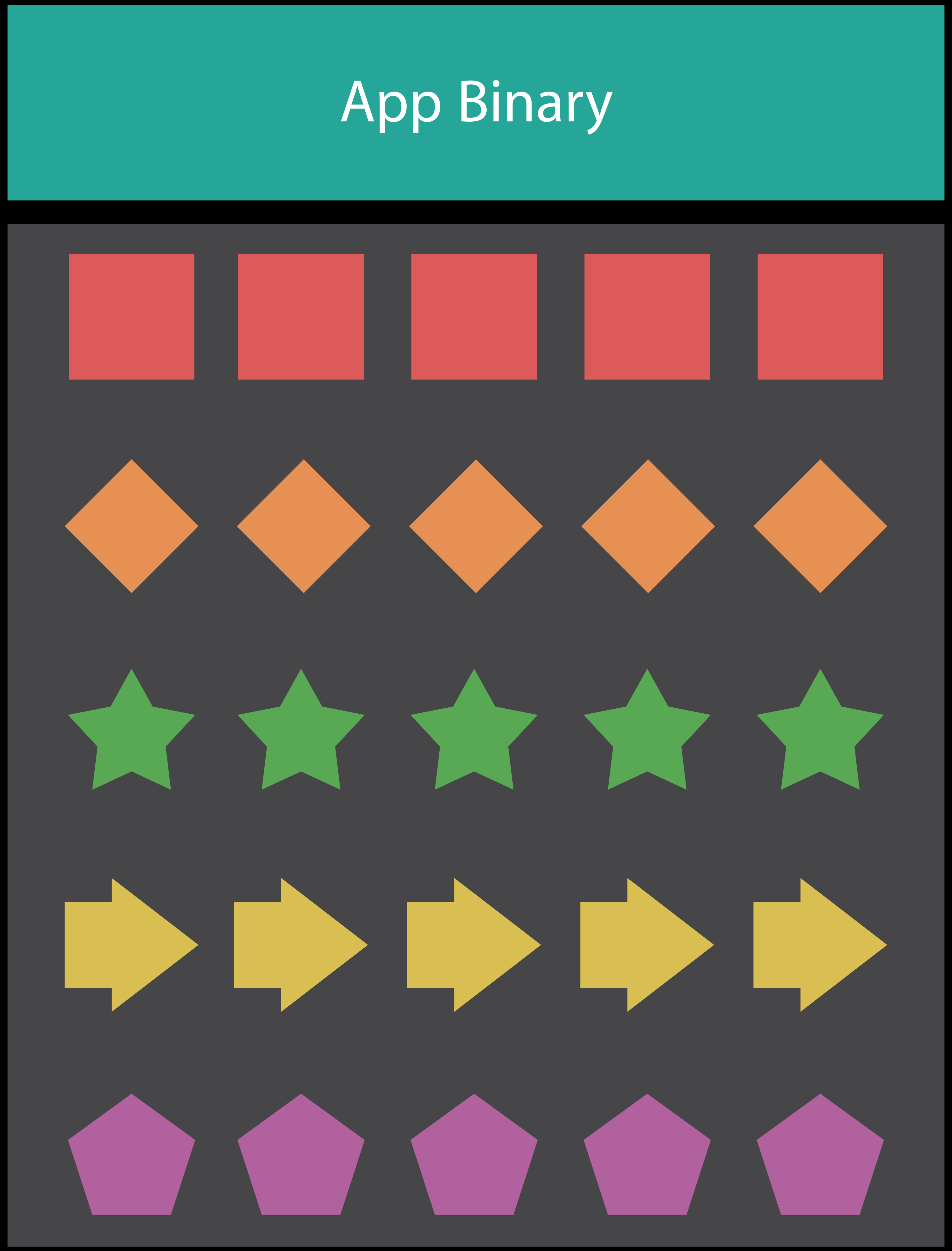
Resources

GreatGame.app



Resources

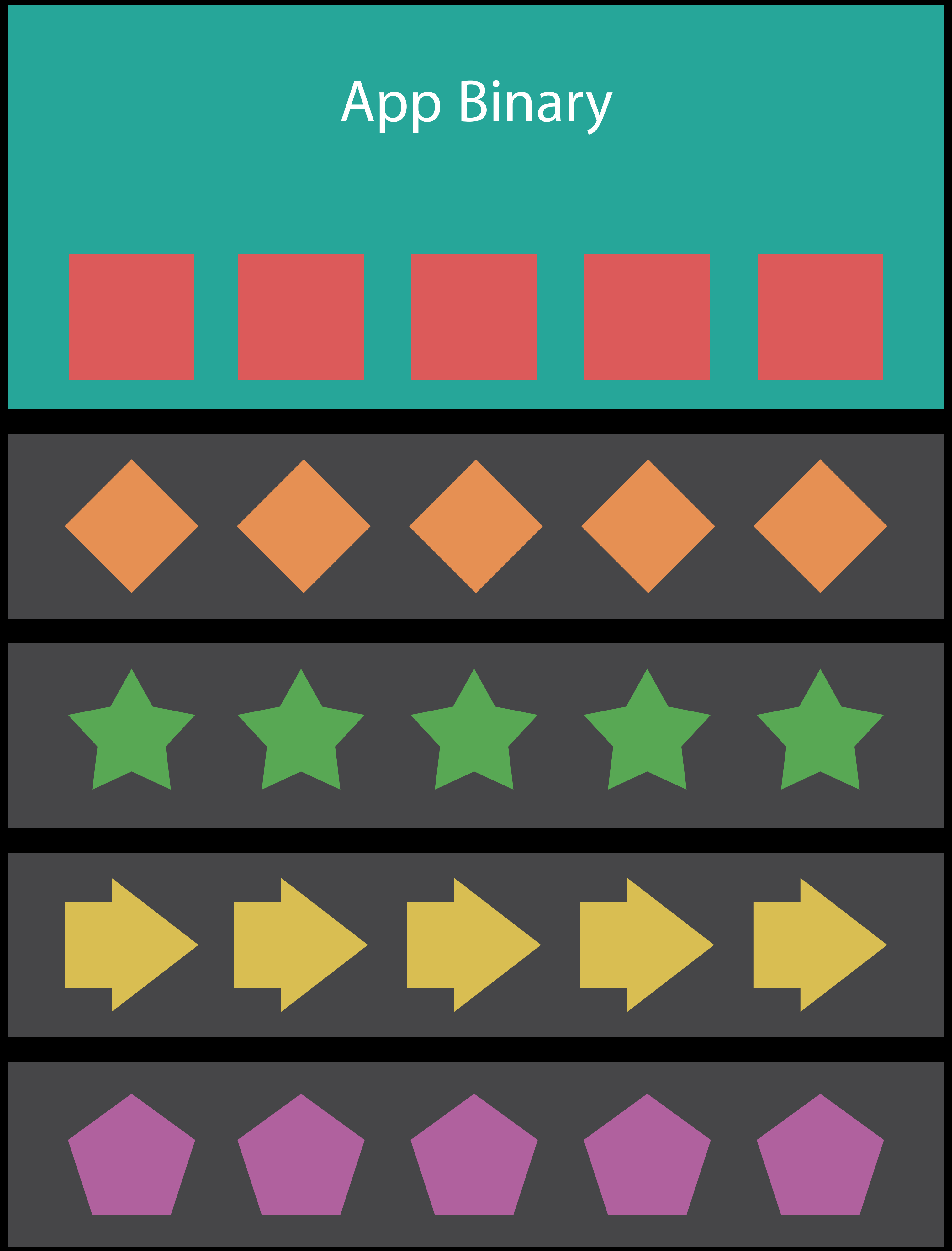
GreatGame.app



Required Always	Include in App Bundle
Level One	“Level 01”
Level Two	“Level 02”
Level Three	“Level 03”
Purchasable Item	“Item 01”

Resources

GreatGame.app



Required Always	Include in App Bundle
Level One	“Level 01”
Level Two	“Level 02”
Level Three	“Level 03”
Purchasable Item	“Item 01”

Assigning Tags
Getting started

Only bundle what's always required

- Assets required throughout app
- UI elements required every launch

Assigning Tags

Getting started

Only bundle what's always required

- Assets required throughout app
- UI elements required every launch

Apply tags to the rest

- Up to 512MB per tag (64MB recommended)
- More than one tag per resource, if needed

Using Tagged Resources

BundleResourceRequest

Overview

Manages access to on-demand resources

- Set up with tags and other options
- Begin and end accessing resources
- Set priority, track progress, handle errors

BundleResourceRequest

Overview

Manages access to on-demand resources

- Set up with tags and other options
- Begin and end accessing resources
- Set priority, track progress, handle errors

Create as many as you need

- Each request is one-shot

BundleResourceRequest

Overview

Manages access to on-demand resources

- Set up with tags and other options
- Begin and end accessing resources
- Set priority, track progress, handle errors

Create as many as you need

- Each request is one-shot

Request decoupled from use of resources

BundleResourceRequest

Core methods

BundleResourceRequest

Core methods

Initialize with set of tags

```
let request = BundleResourceRequest(tags: [“Level1”])
```

BundleResourceRequest

Core methods

Initialize with set of tags

```
let request = BundleResourceRequest(tags: [“Level1”])
```

Begin a request

```
request.beginAccessingResources { (error: NSError?) in

    …

}
```

BundleResourceRequest

Core methods

Initialize with set of tags

```
let request = BundleResourceRequest(tags: [“Level1”])
```

Begin a request

```
request.beginAccessingResources { (error: NSError?) in

    …

}
```

Access resources

```
var mapPath = request.bundle.pathForResource("Level1", ofType: "map")
```

BundleResourceRequest

Core methods

Initialize with set of tags

```
let request = BundleResourceRequest(tags: [“Level1”])
```

Begin a request

```
request.beginAccessingResources { (error: NSError?) in
    …
}
```

Access resources

```
var mapPath = request.bundle.pathForResource("Level1", ofType: "map")
```

Tell the system you’re finished

```
request.endAccessingResources()
```

BundleResourceRequest

Loading priority

BundleResourceRequest

Loading priority

Set relative priority of simultaneous requests

```
        var loadingPriority: Double
```

- Value ranges from 0.0 to 1.0

BundleResourceRequest

Loading priority

Set relative priority of simultaneous requests

```
var loadingPriority: Double
```

- Value ranges from 0.0 to 1.0

Special urgent priority

```
request.loadingPriority = NSBundleResourceRequestLoadingPriorityUrgent
```

- Suspends other downloads
- Maximized throughput (and CPU consumption)

BundleResourceRequest

Conditional requests

BundleResourceRequest

Conditional requests

To check if content is present

```
request.conditionallyBeginAccessingResources {  (available: Bool) in

    …

}
```

BundleResourceRequest

Conditional requests

To check if content is present

```
request.conditionallyBeginAccessingResources {  (available: Bool) in

    …

}
```

If already downloaded, then identical to `beginAccessingResources()`

BundleResourceRequest

Conditional requests

To check if content is present

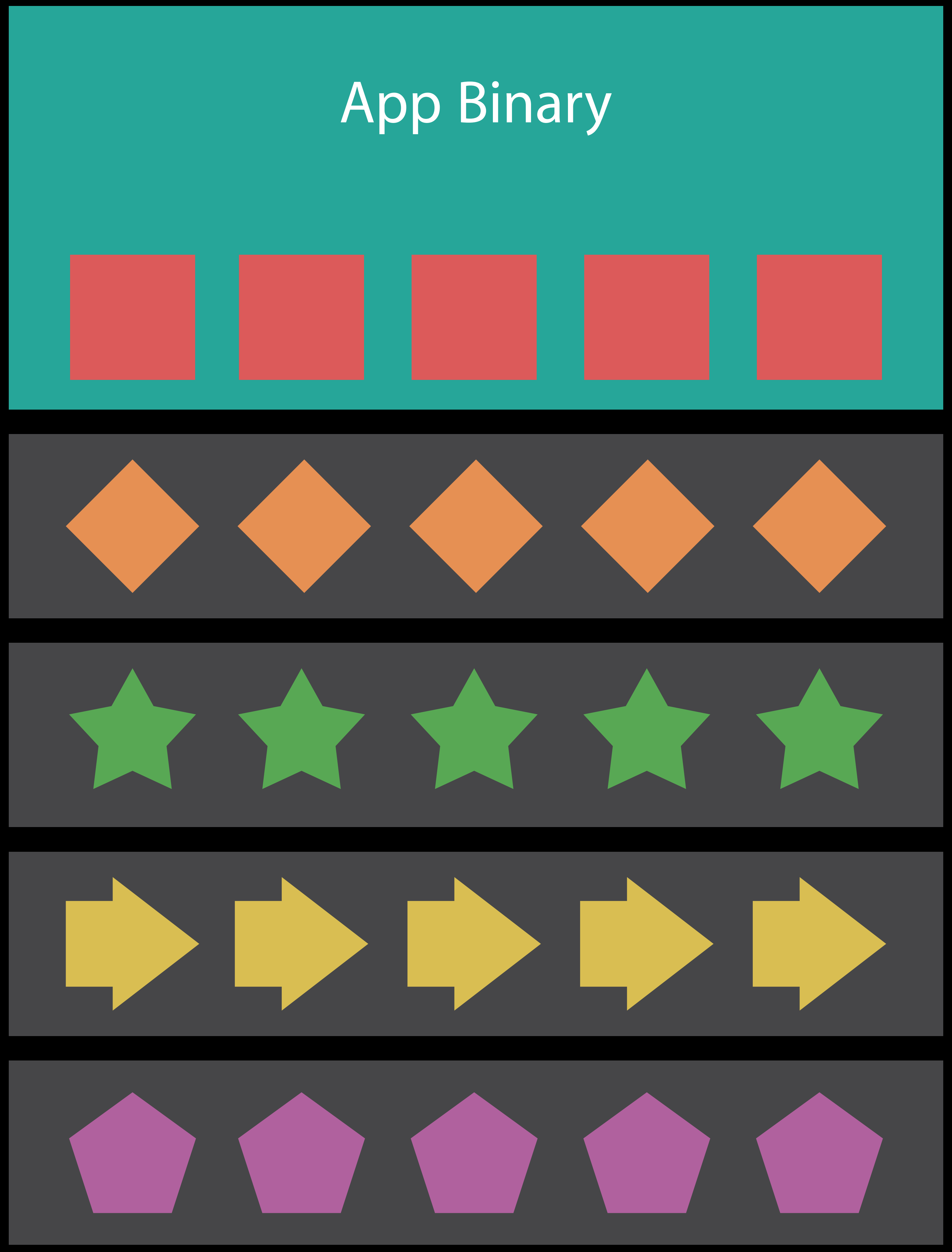
```
request.conditionallyBeginAccessingResources {  (available: Bool) in
    …
}
```

If already downloaded, then identical to `beginAccessingResources()`

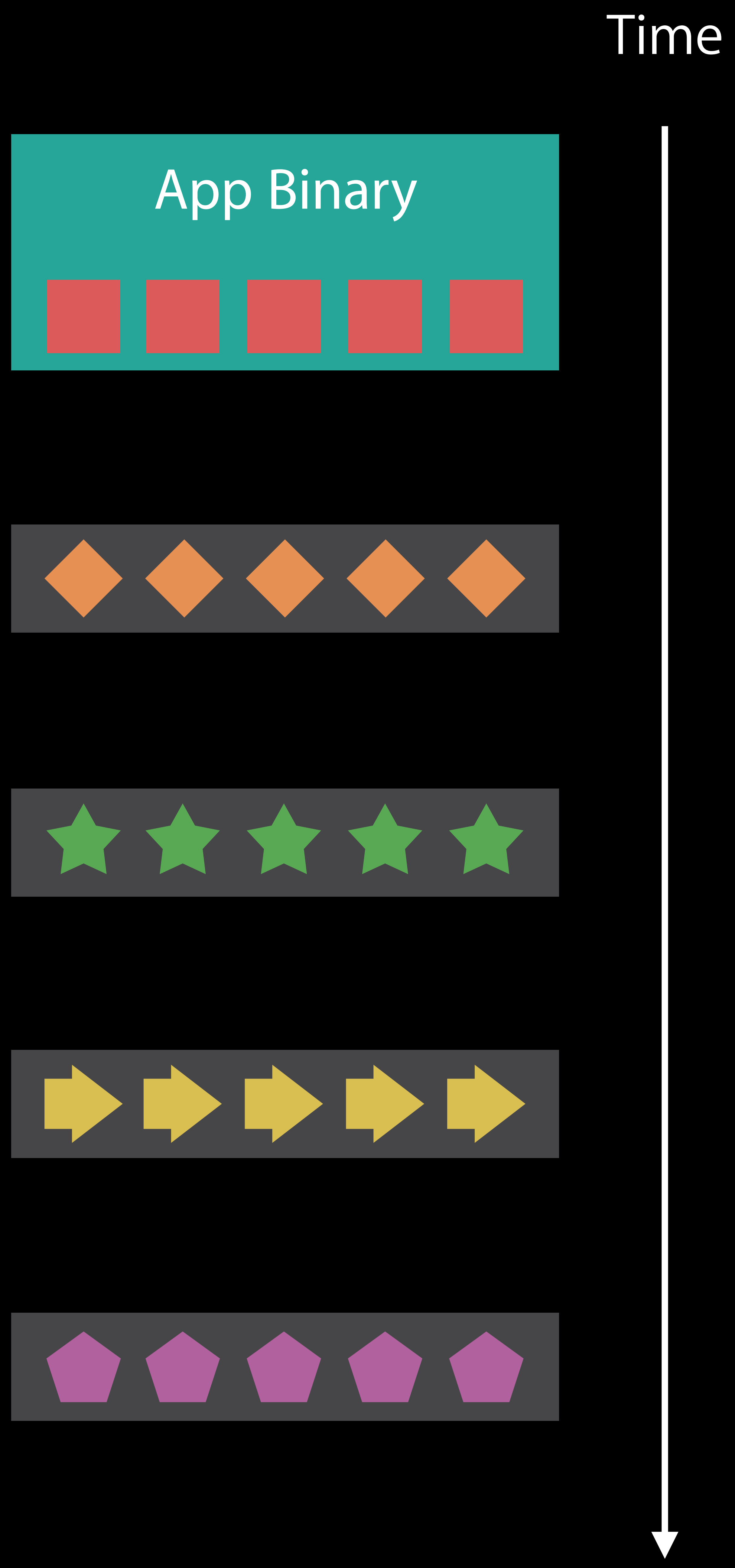
Always call `endAccessingResources()`

Optimizing First Launch

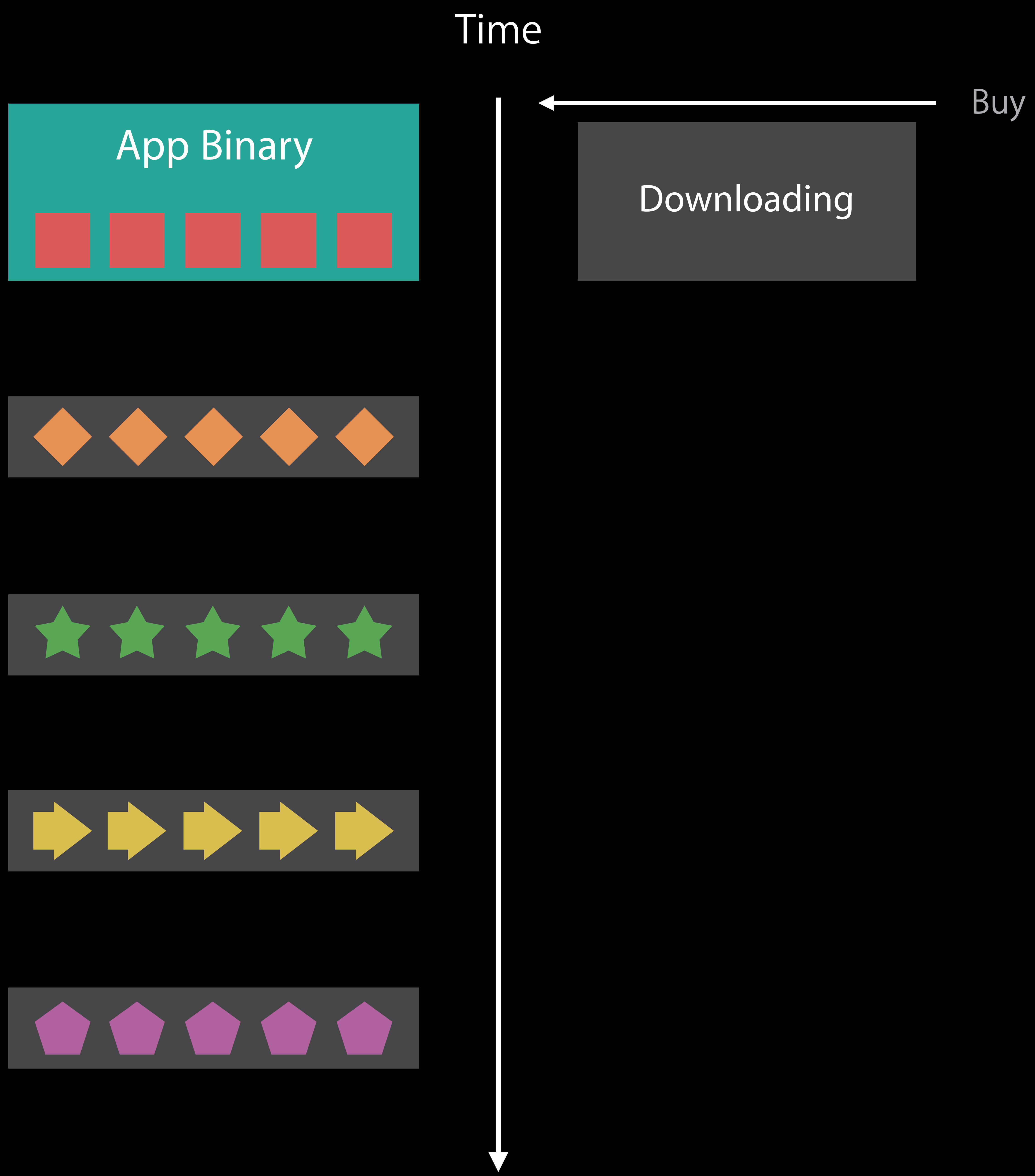
Launch Timeline



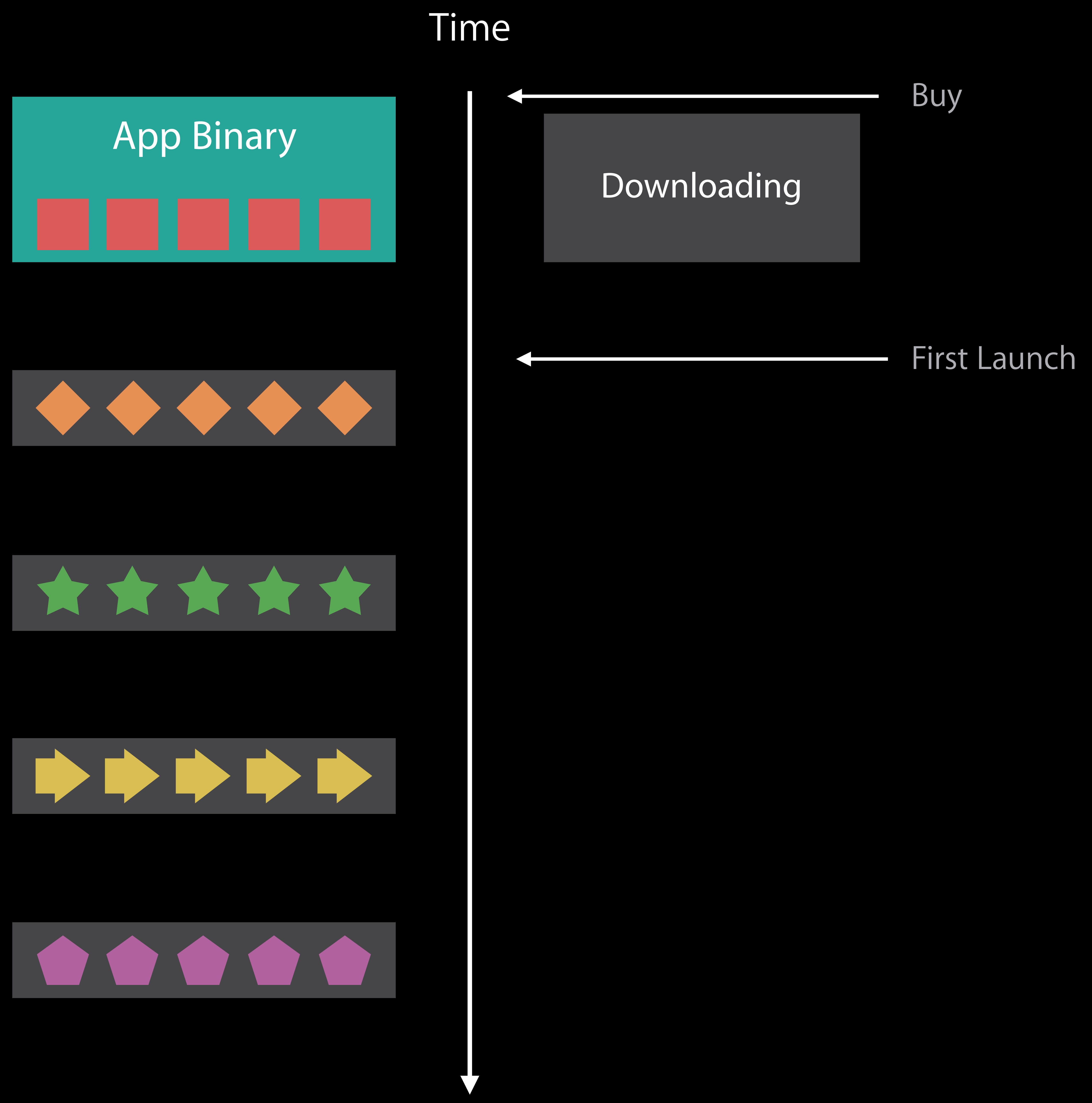
Launch Timeline



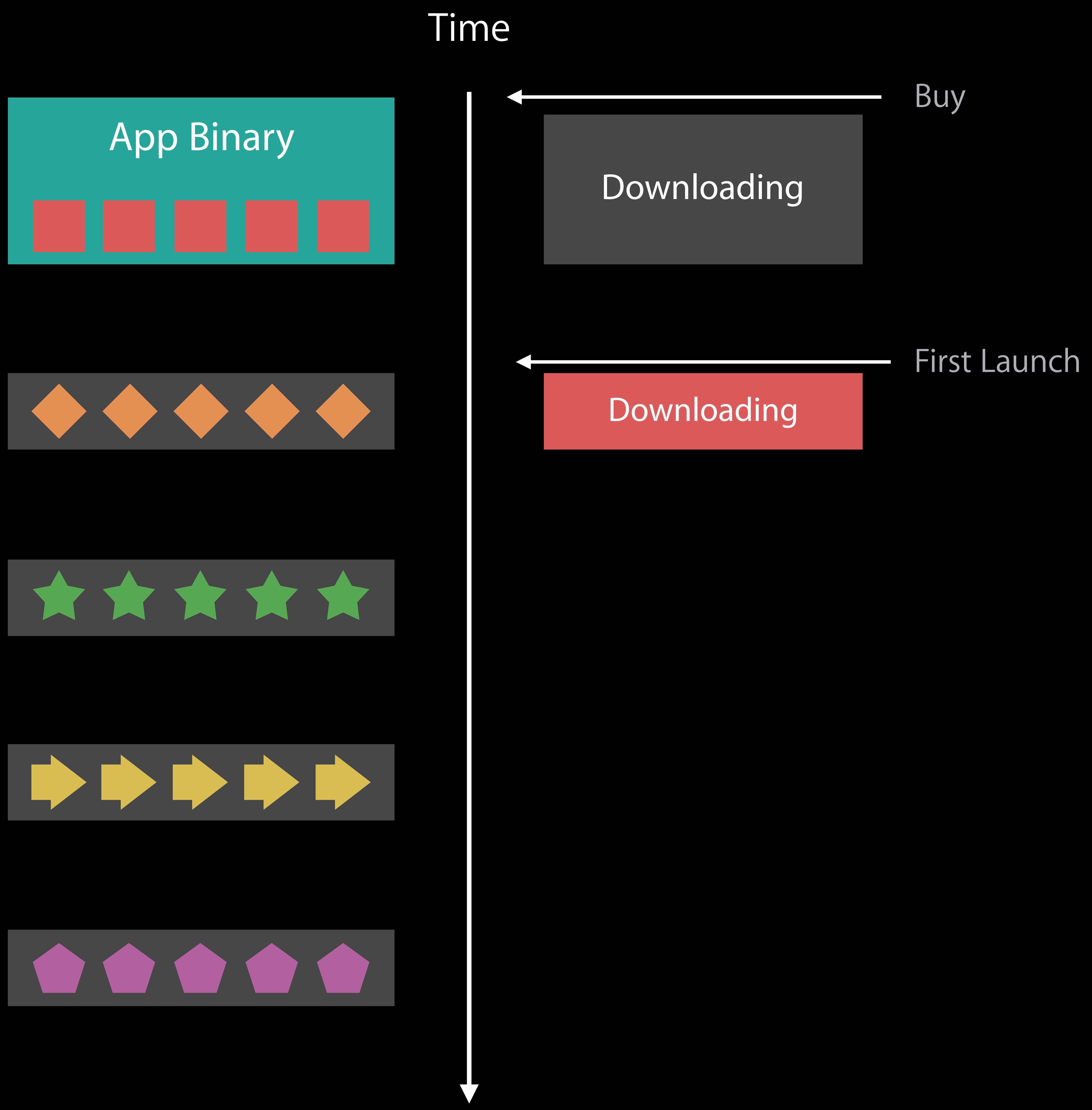
Launch Timeline



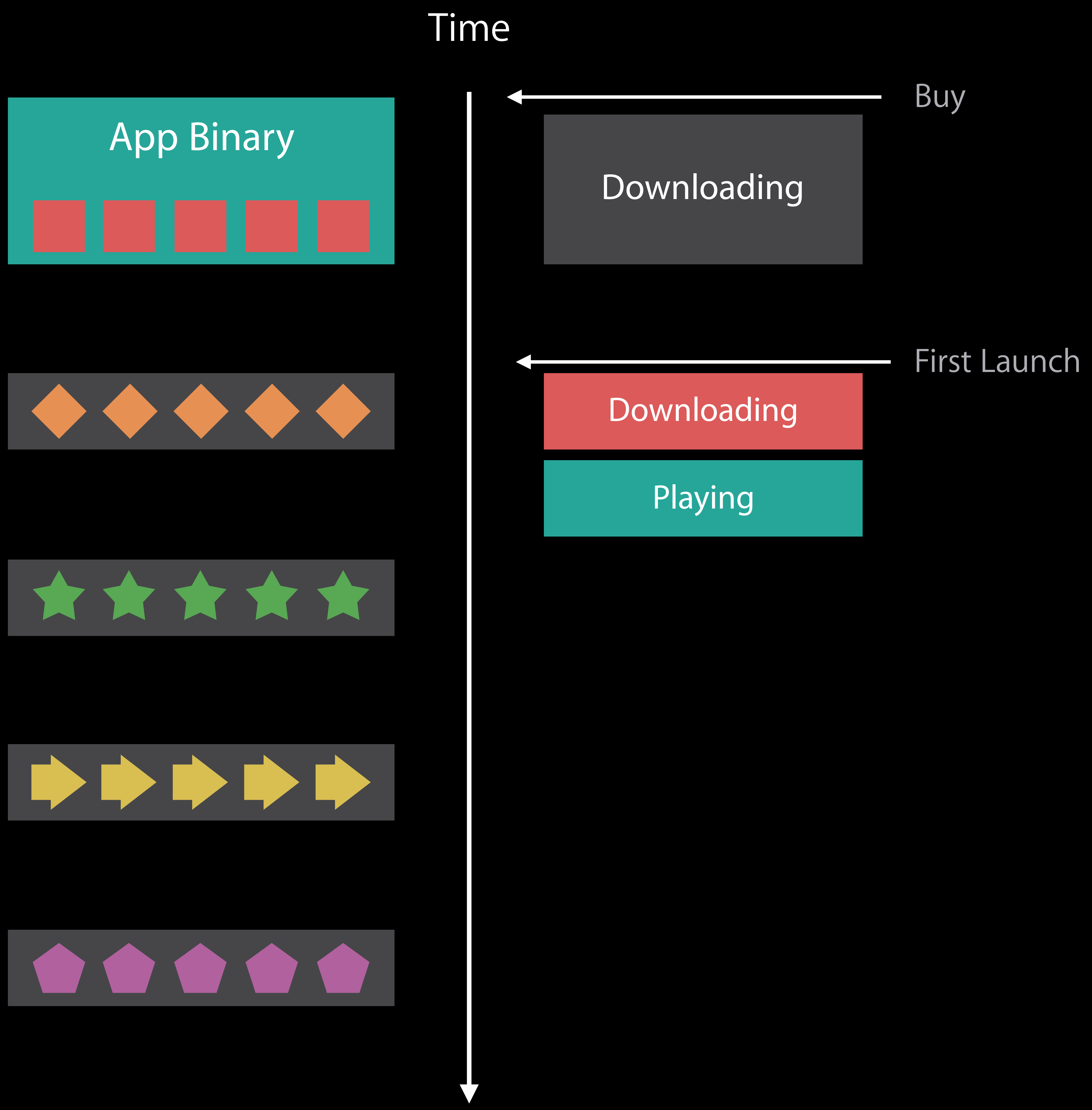
Launch Timeline



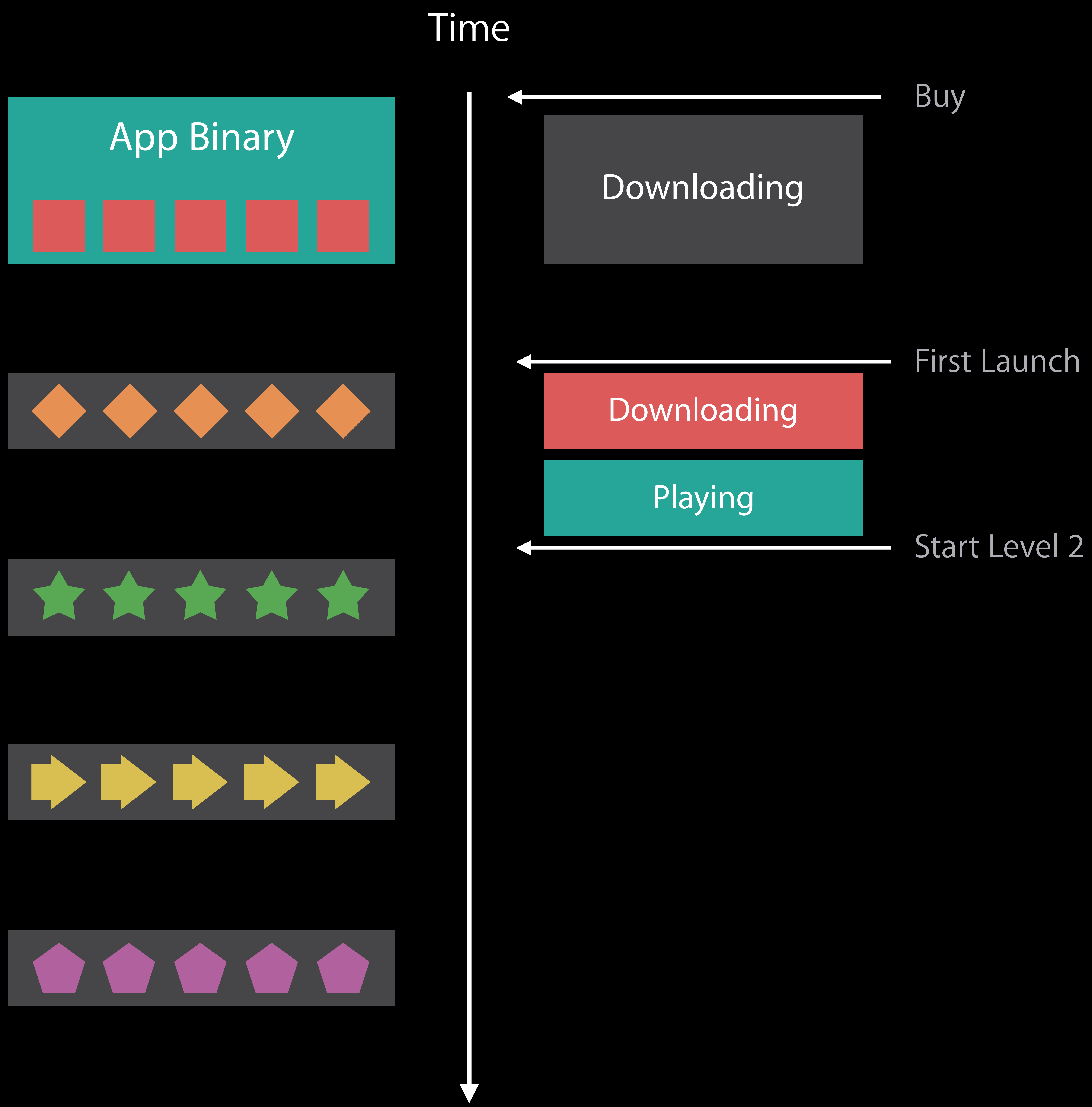
Launch Timeline



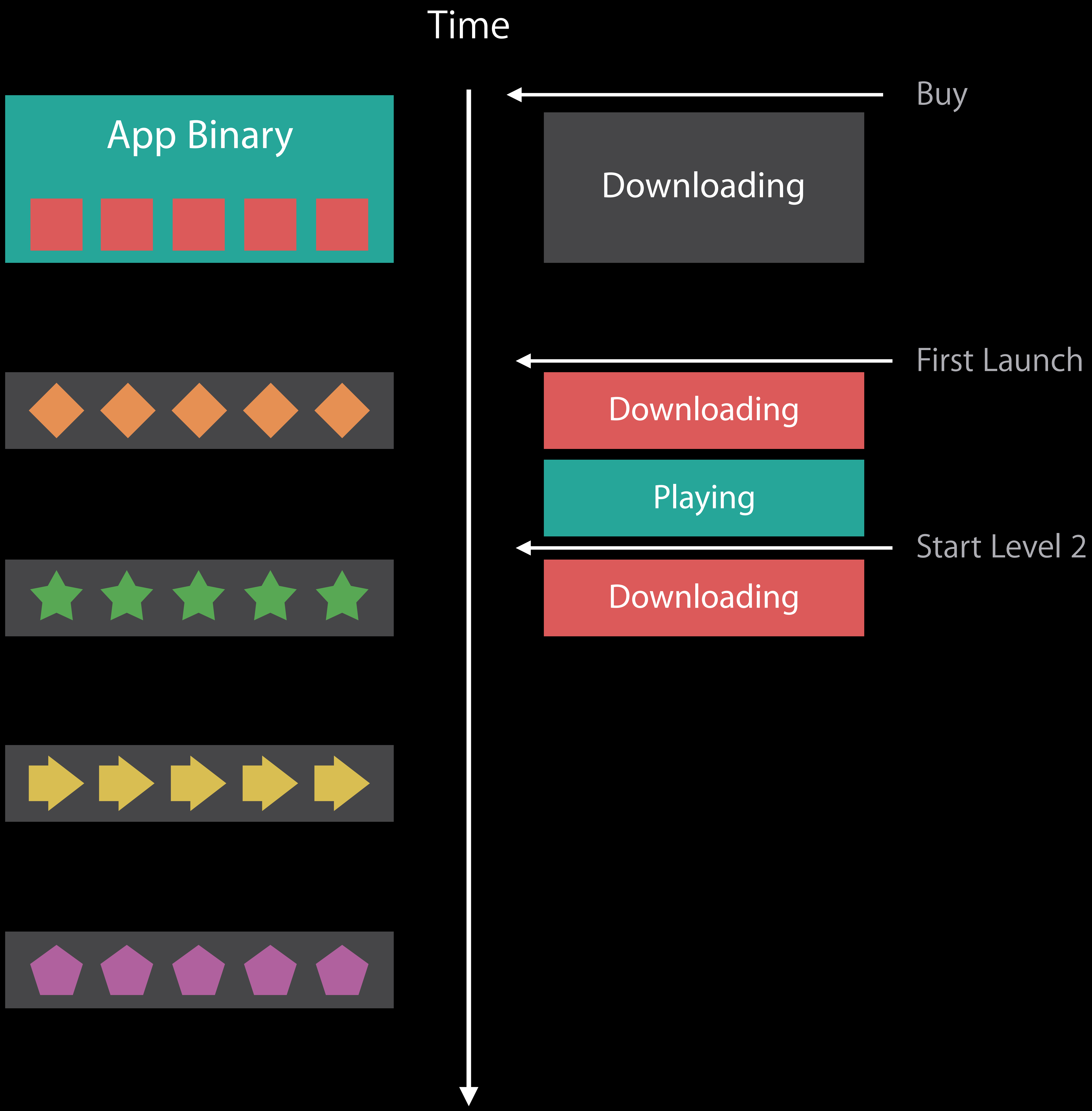
Launch Timeline



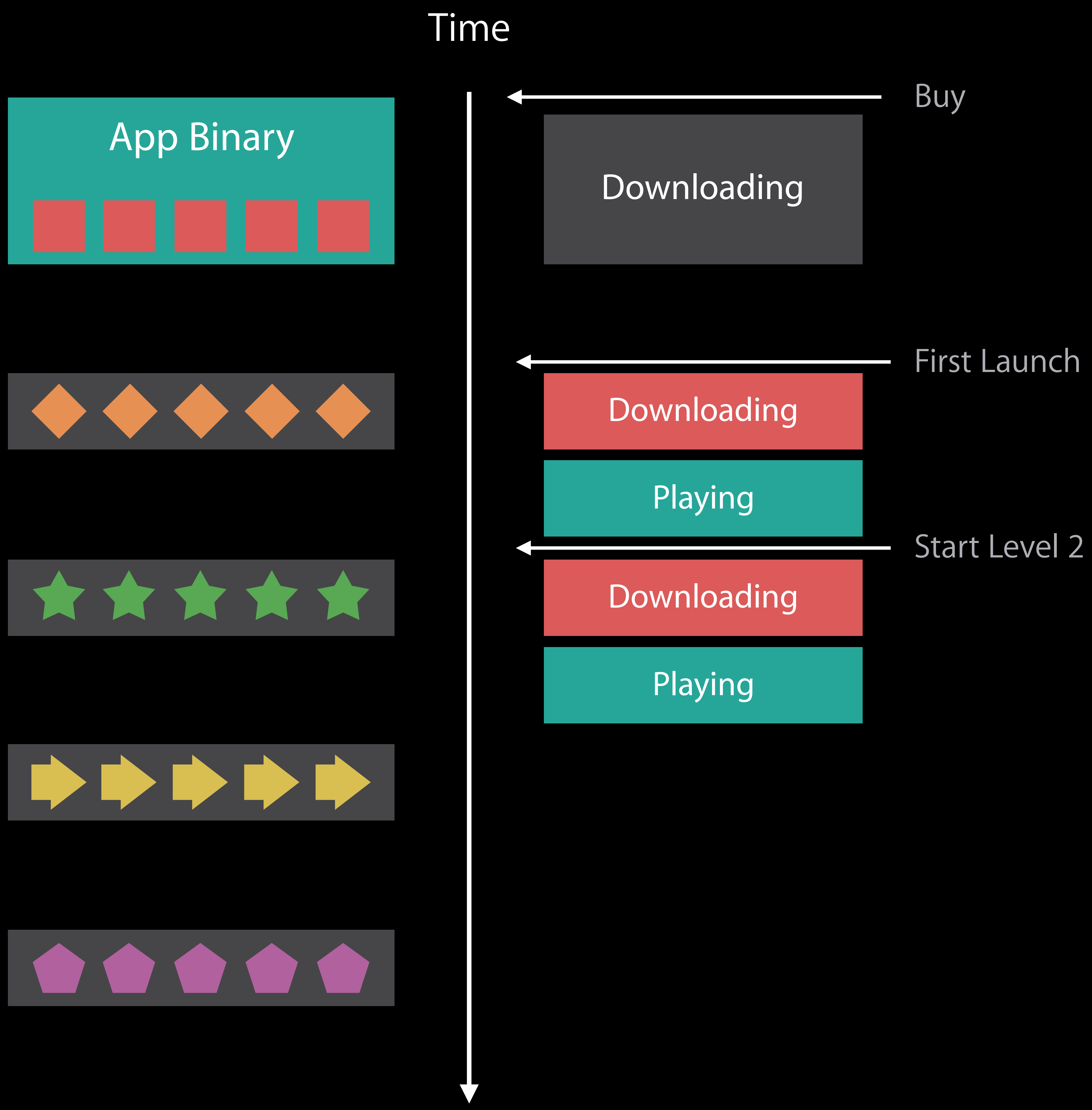
Launch Timeline



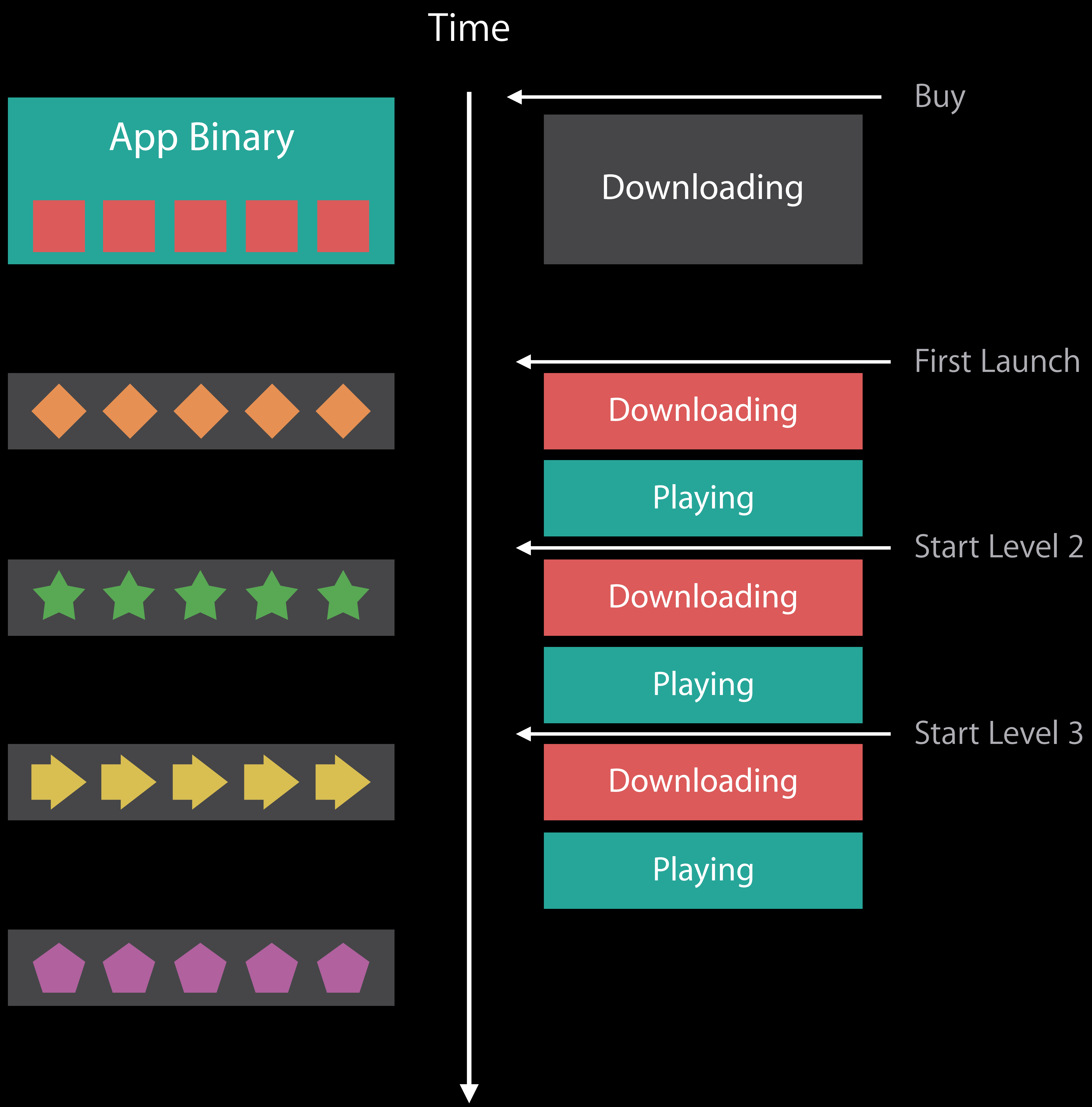
Launch Timeline



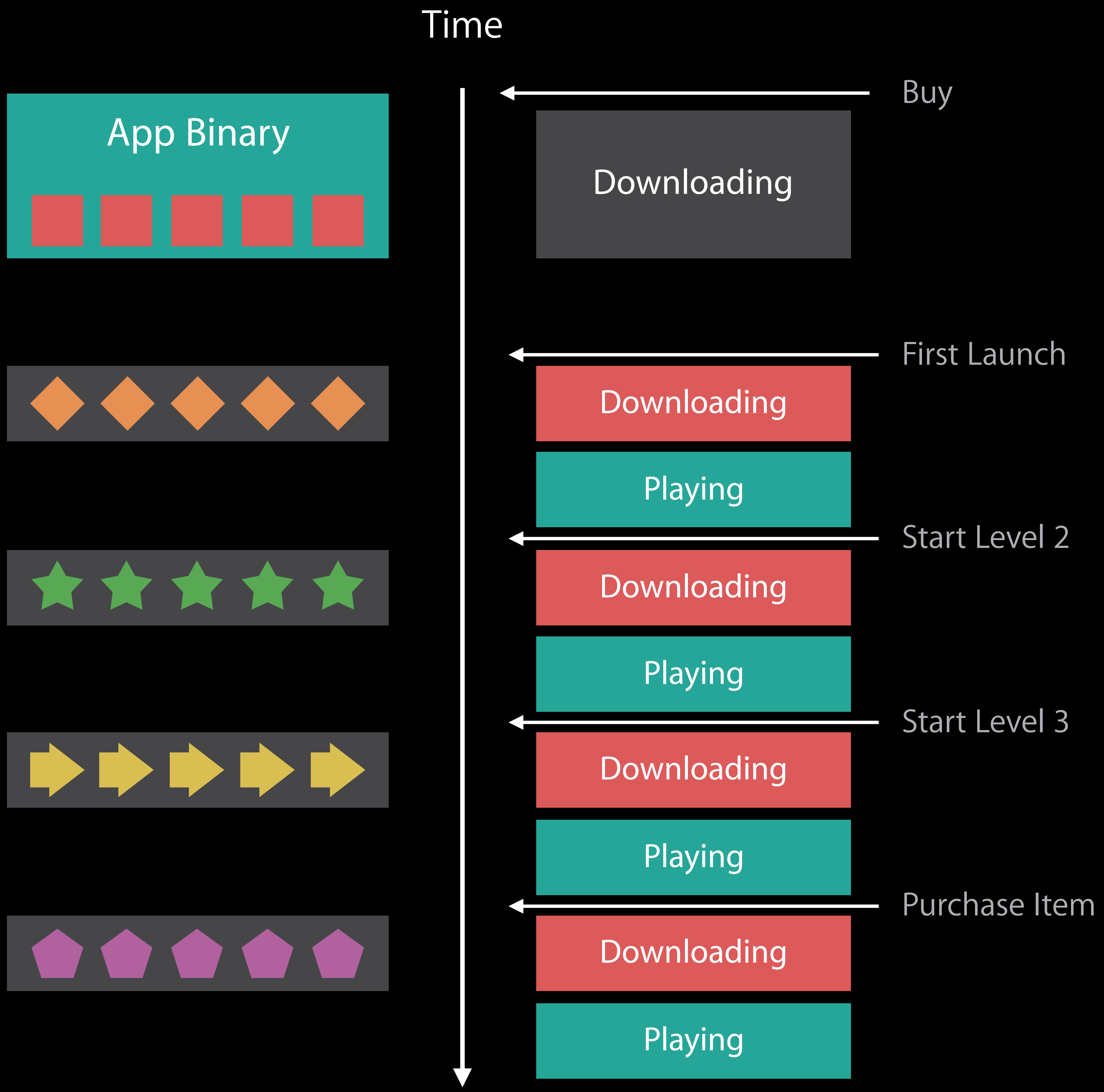
Launch Timeline



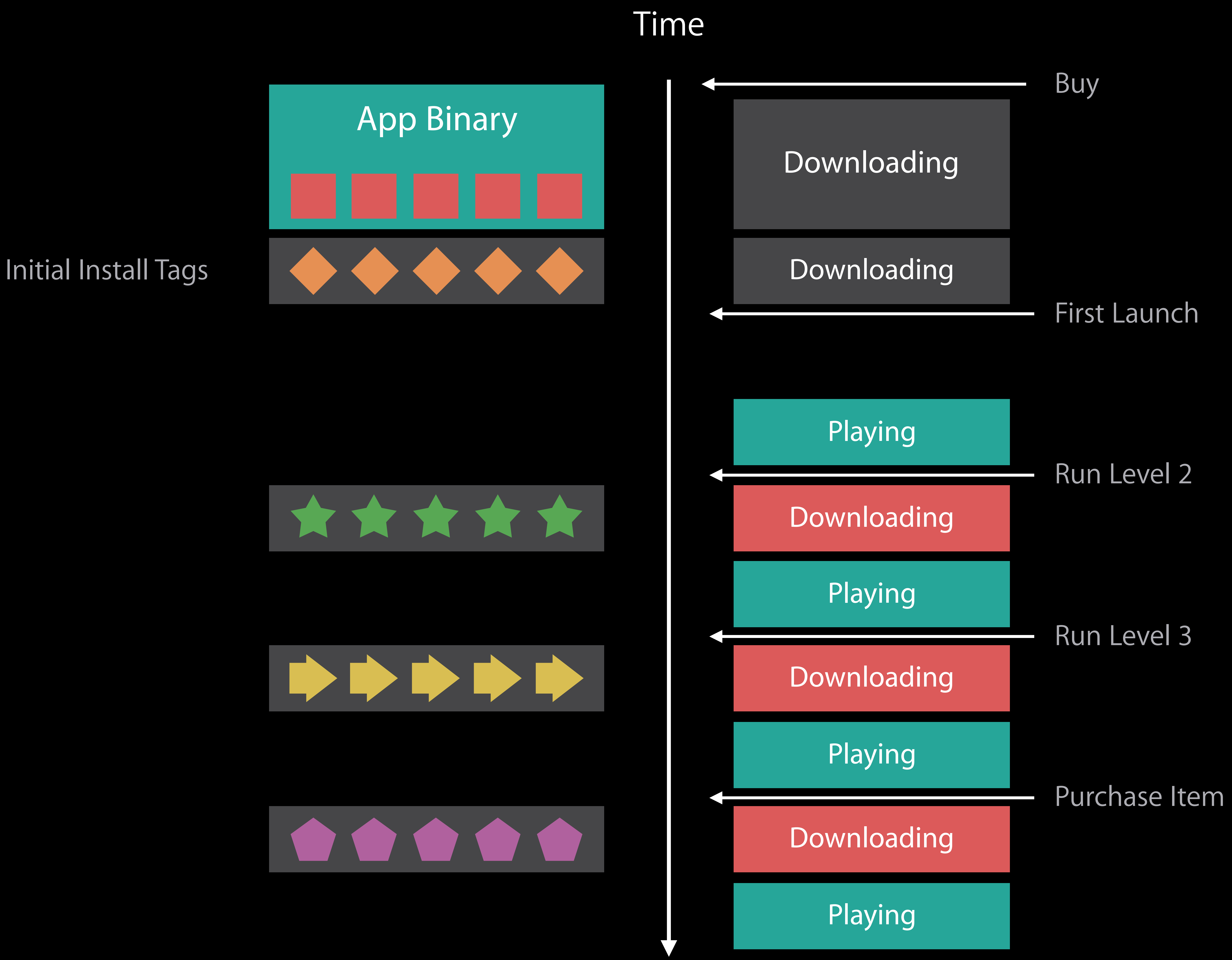
Launch Timeline



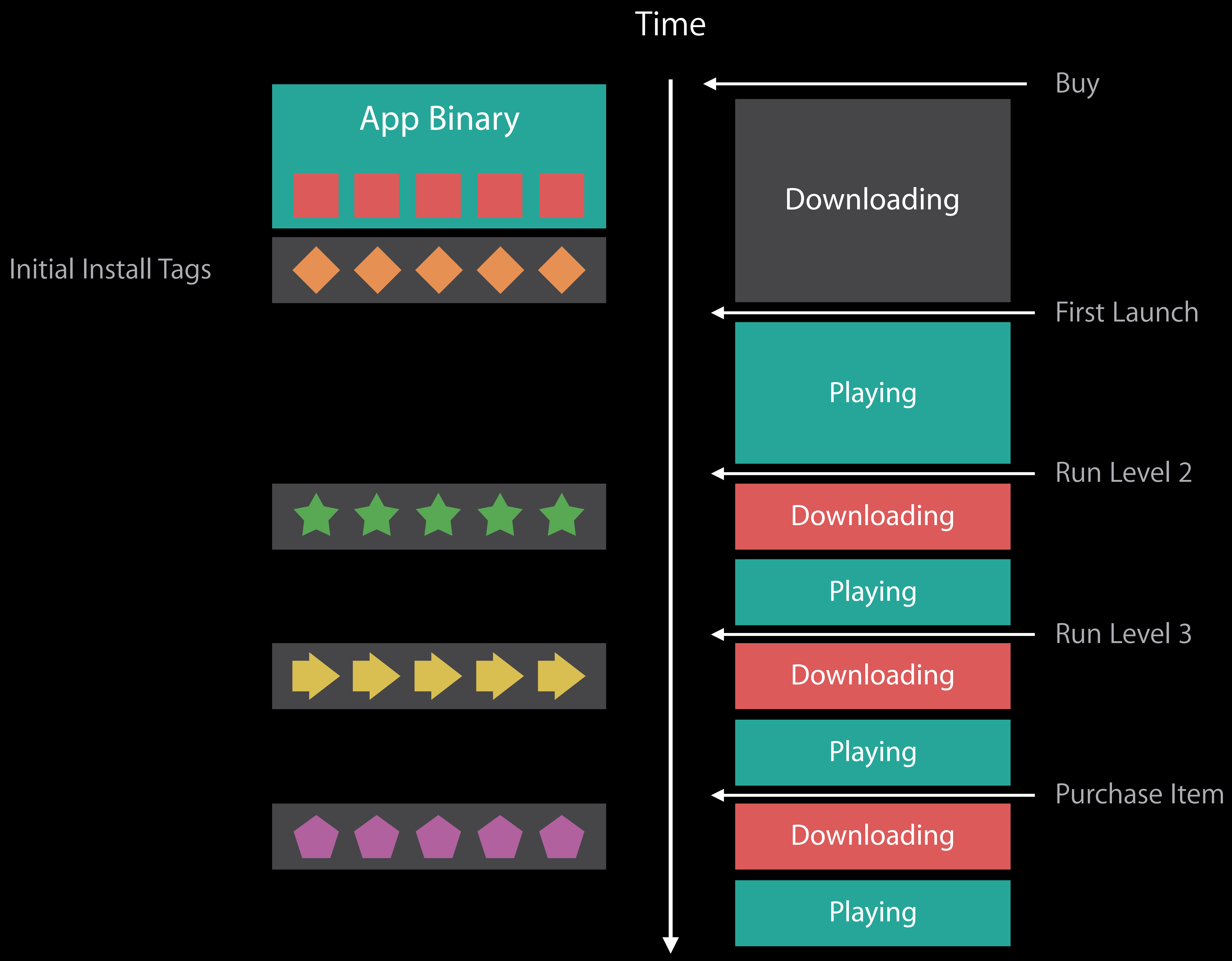
Launch Timeline



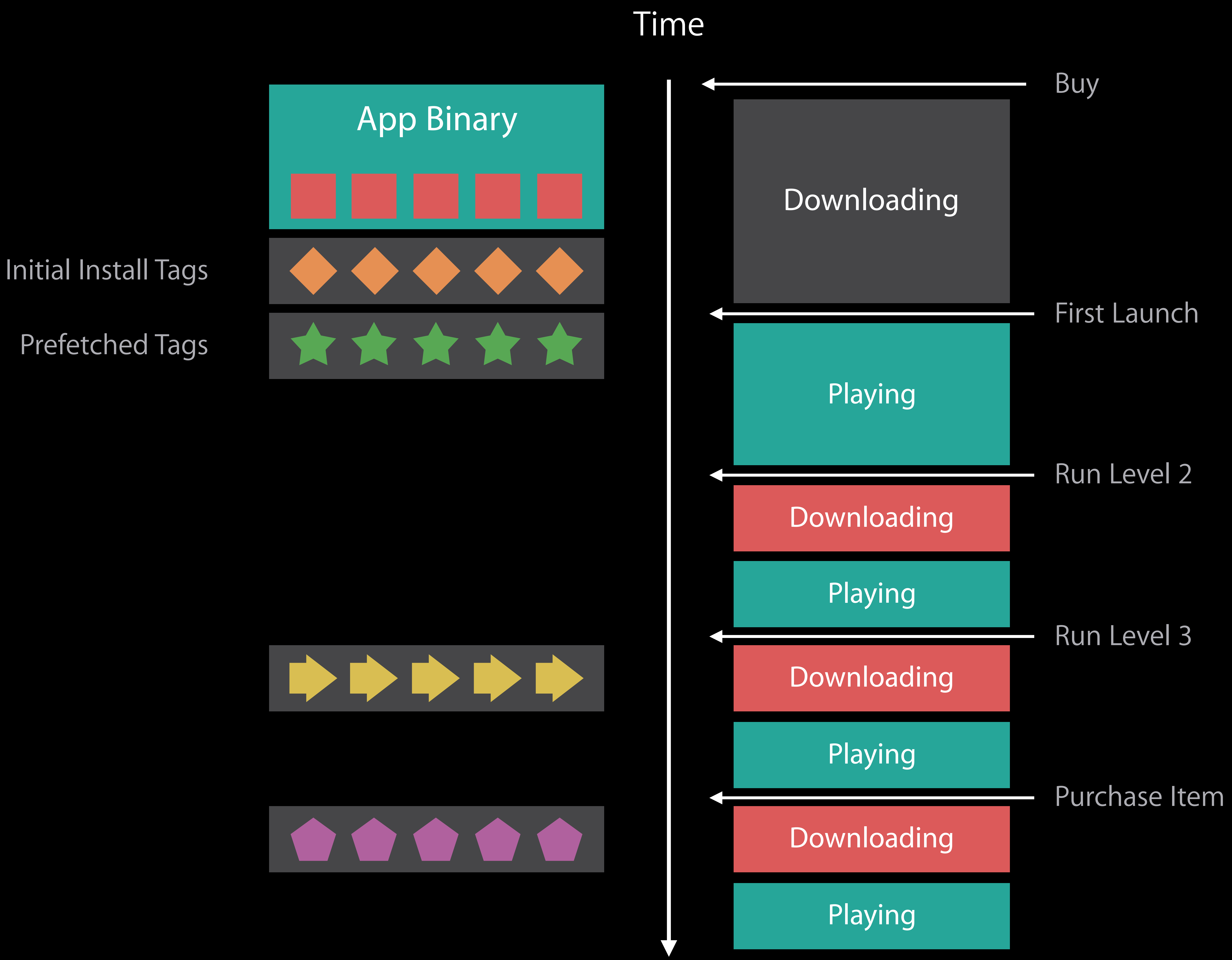
Launch Timeline



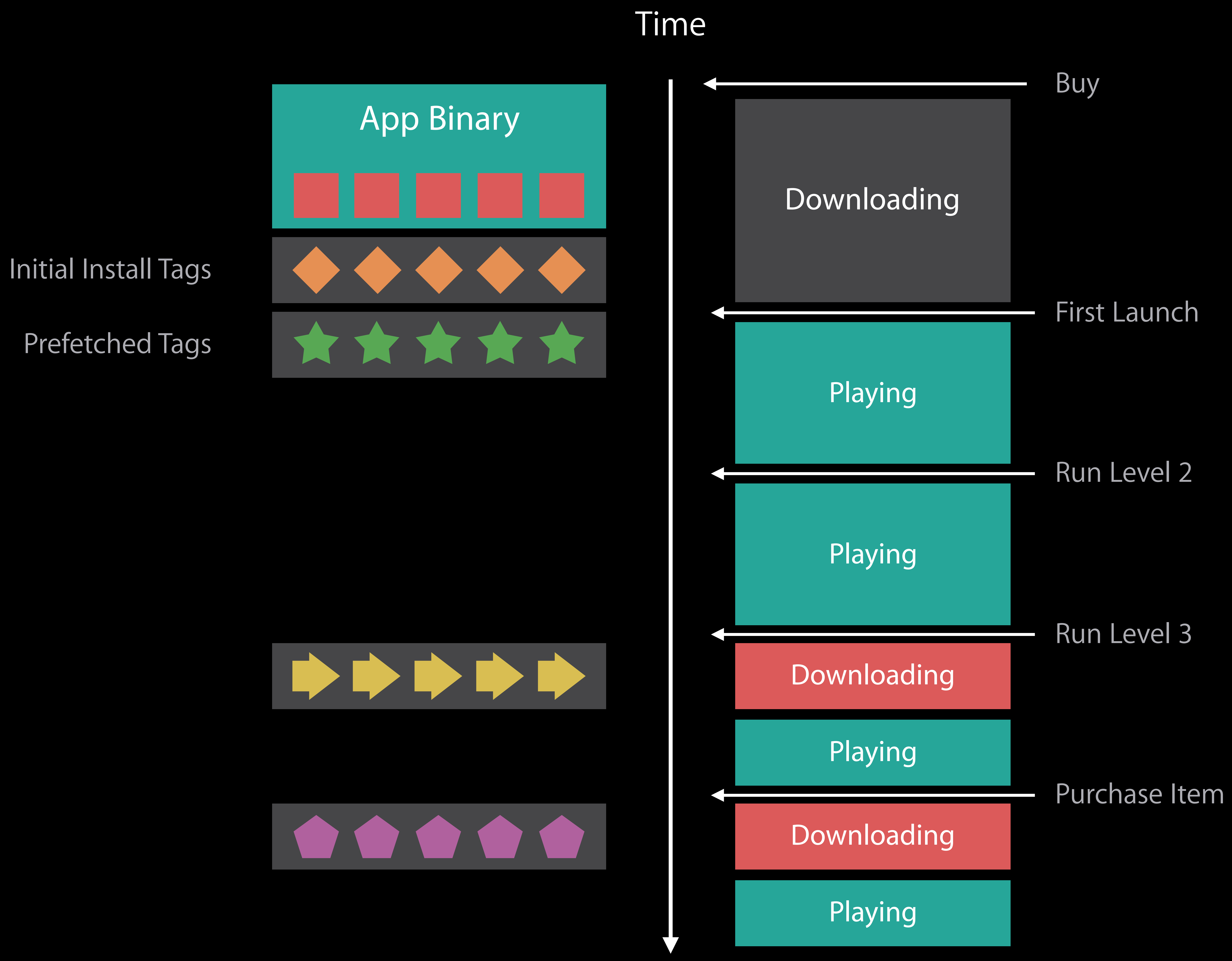
Launch Timeline



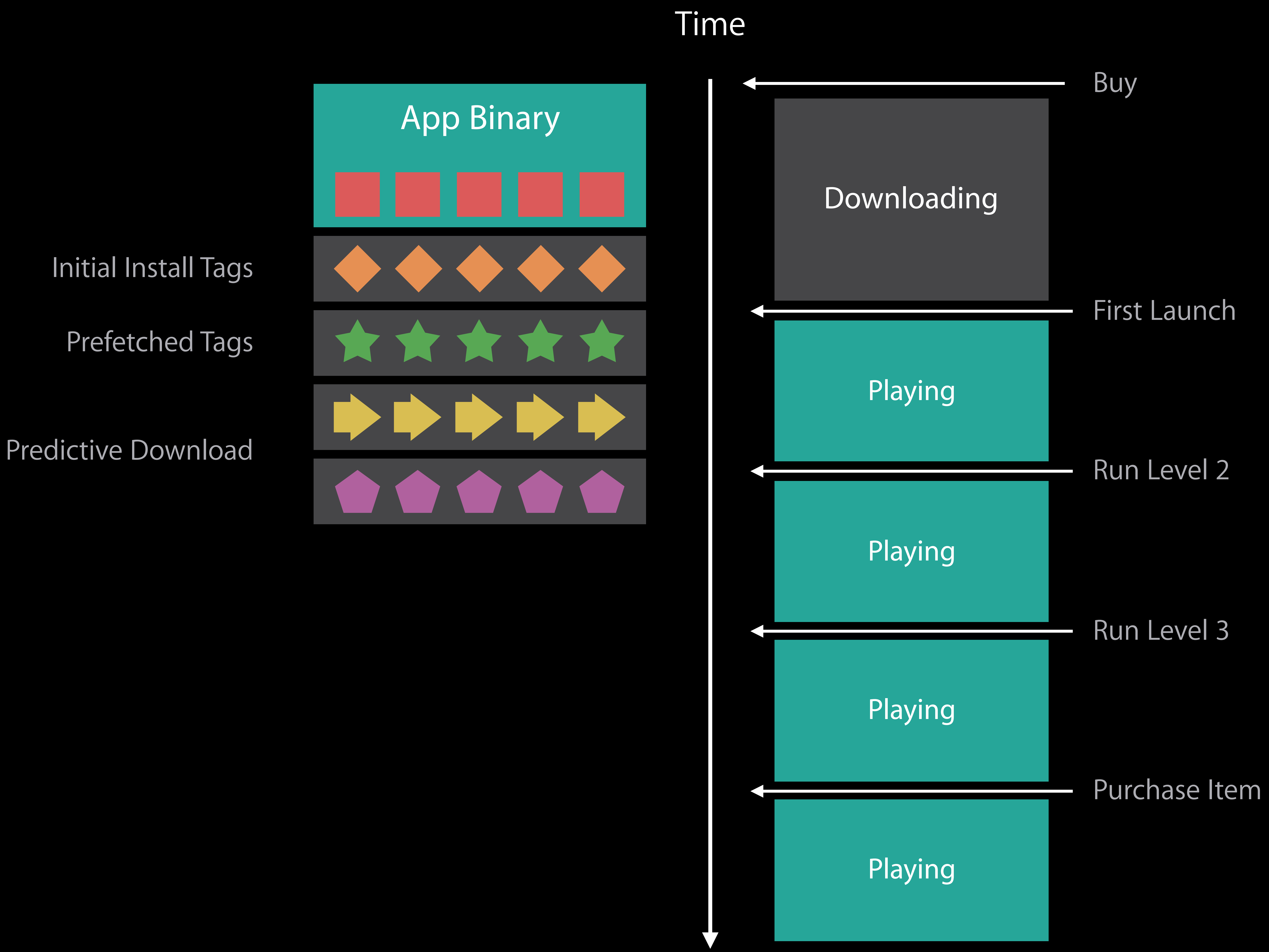
Launch Timeline



Launch Timeline



Launch Timeline



Optimizing First Launch

Initial install and prefetched resources

Initial Install Tags

- Downloaded and installed with the app
- Up to 2GB
- Part of the "Size" shown in the App Store

Optimizing First Launch

Initial install and prefetched resources

Initial Install Tags

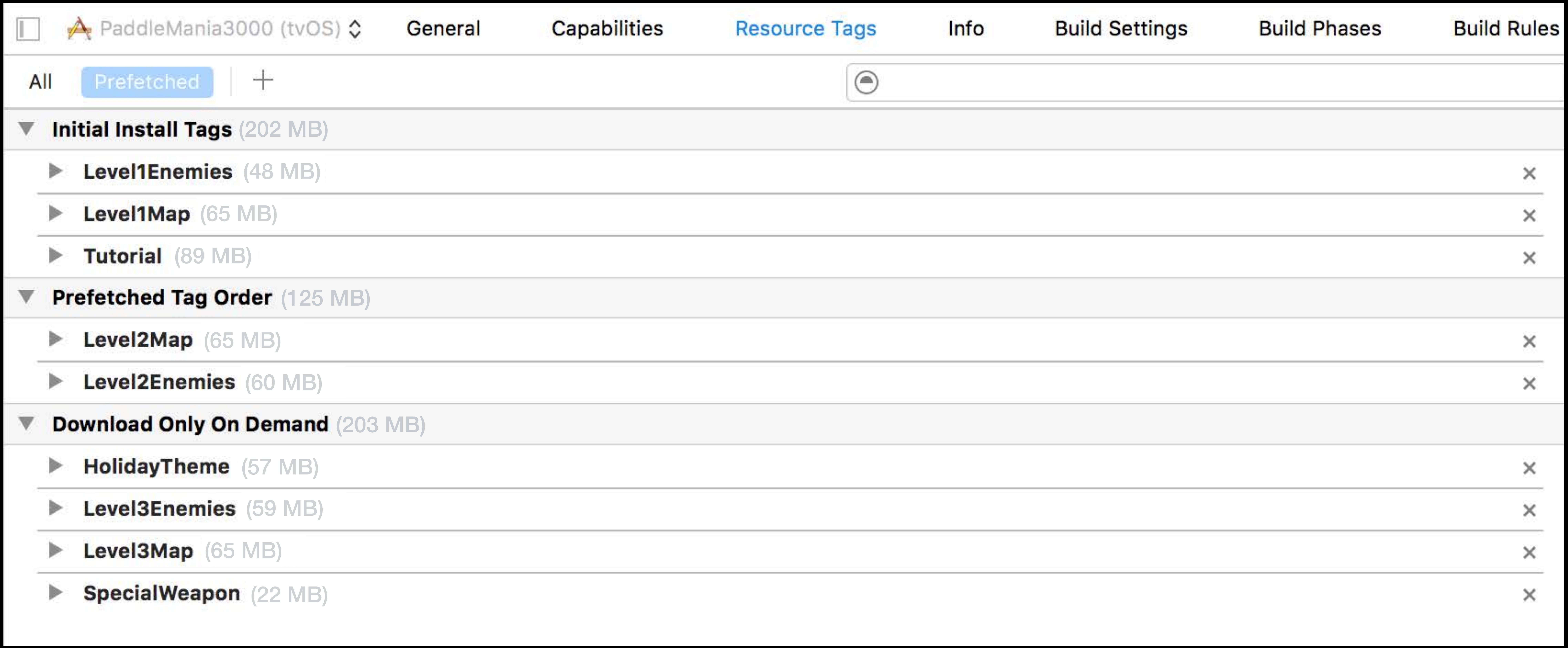
- Downloaded and installed with the app
- Up to 2GB
- Part of the "Size" shown in the App Store

Prefetched tag order

- Automatically prefetches after app download
- Up to 4GB – sizeof(Initial Install Tags)
- Follows order specified in Xcode

Optimizing First Launch

Initial install and prefetched resources



PaddleMania3000 (tvOS) | General | Capabilities | Resource Tags | Info | Build Settings | Build Phases | Build Rules

All | Prefetched | +

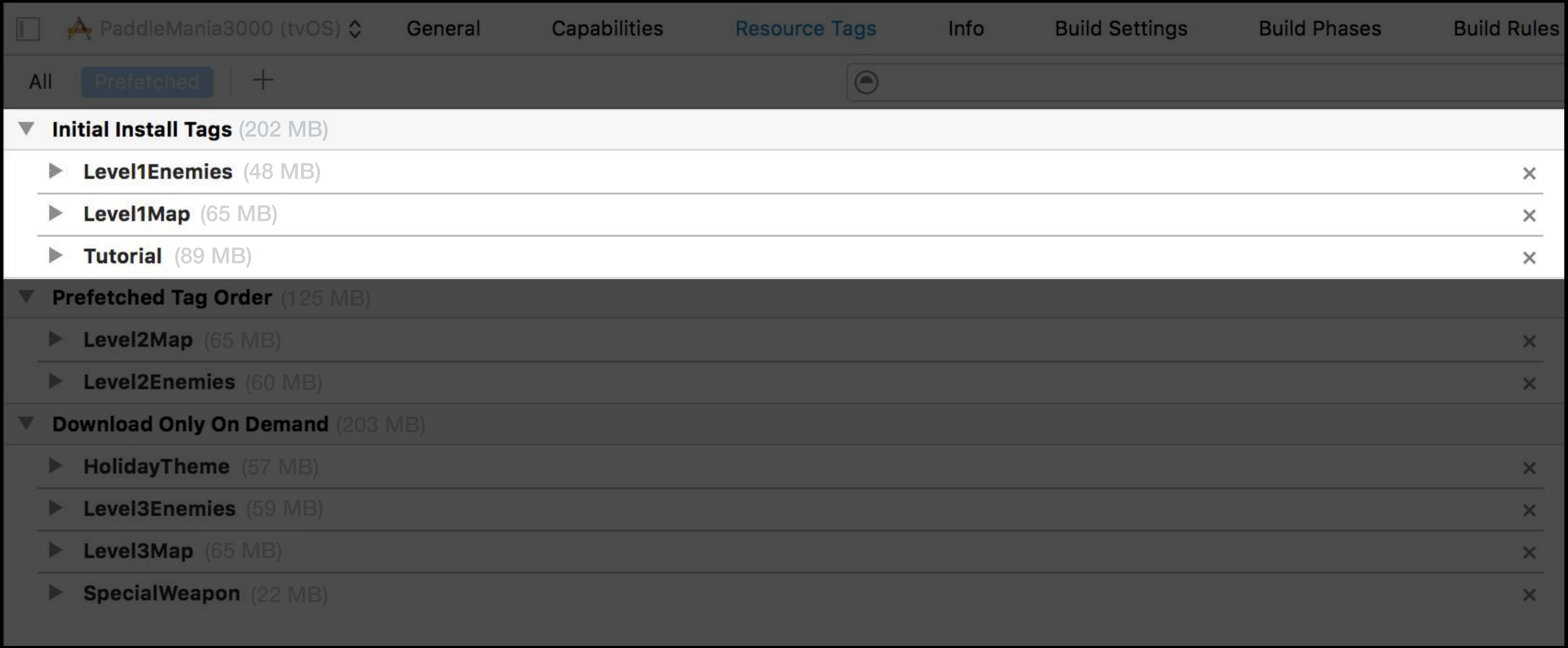
Initial Install Tags (202 MB)
- **Level1Enemies** (48 MB)
- **Level1Map** (65 MB)
- **Tutorial** (89 MB)

Prefetched Tag Order (125 MB)
- **Level2Map** (65 MB)
- **Level2Enemies** (60 MB)

Download Only On Demand (203 MB)
- **HolidayTheme** (57 MB)
- **Level3Enemies** (59 MB)
- **Level3Map** (65 MB)
- **SpecialWeapon** (22 MB)

Optimizing First Launch

Initial install and prefetched resources



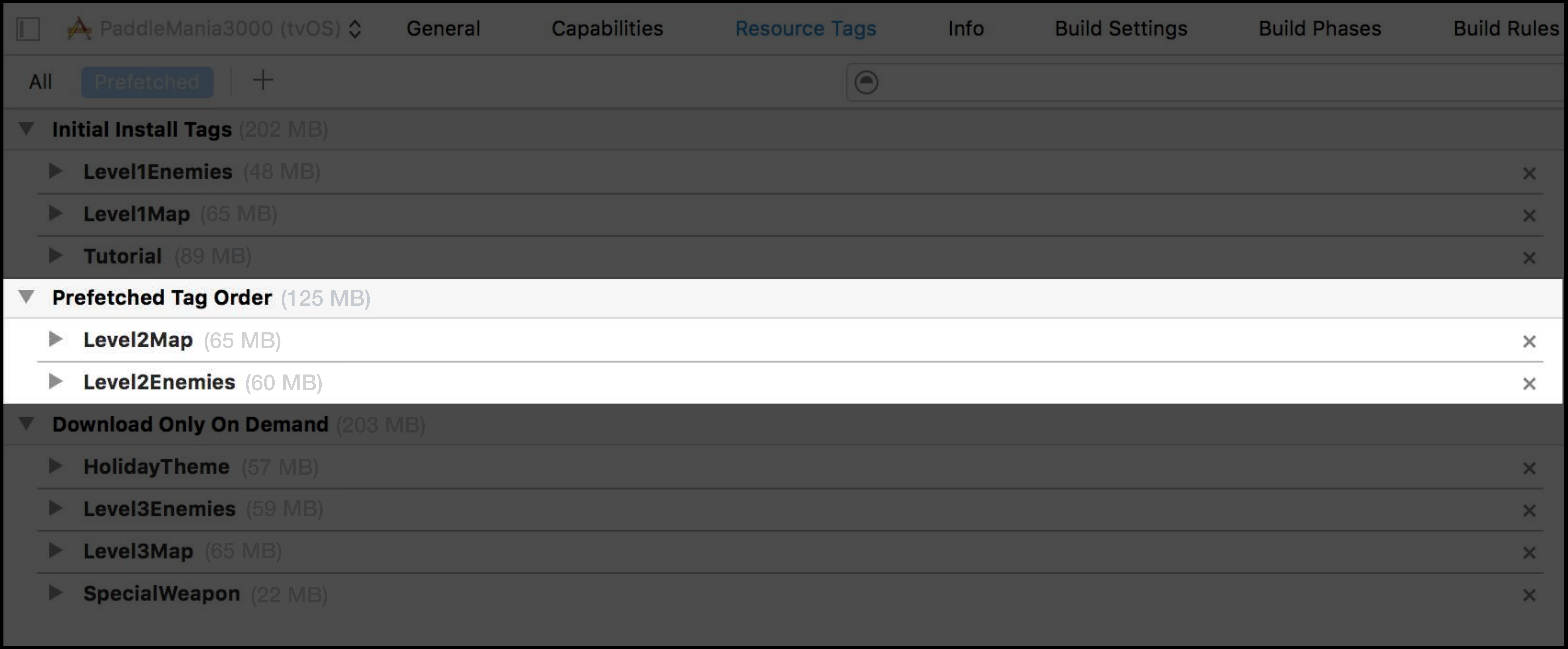
Initial Install Tags (202 MB)
- **Level1Enemies** (48 MB)
- **Level1Map** (65 MB)
- **Tutorial** (89 MB)

Prefetched Tag Order (125 MB)
- **Level2Map** (65 MB)
- **Level2Enemies** (60 MB)

Download Only On Demand (203 MB)
- **HolidayTheme** (57 MB)
- **Level3Enemies** (59 MB)
- **Level3Map** (65 MB)
- **SpecialWeapon** (22 MB)

Optimizing First Launch

Initial install and prefetched resources



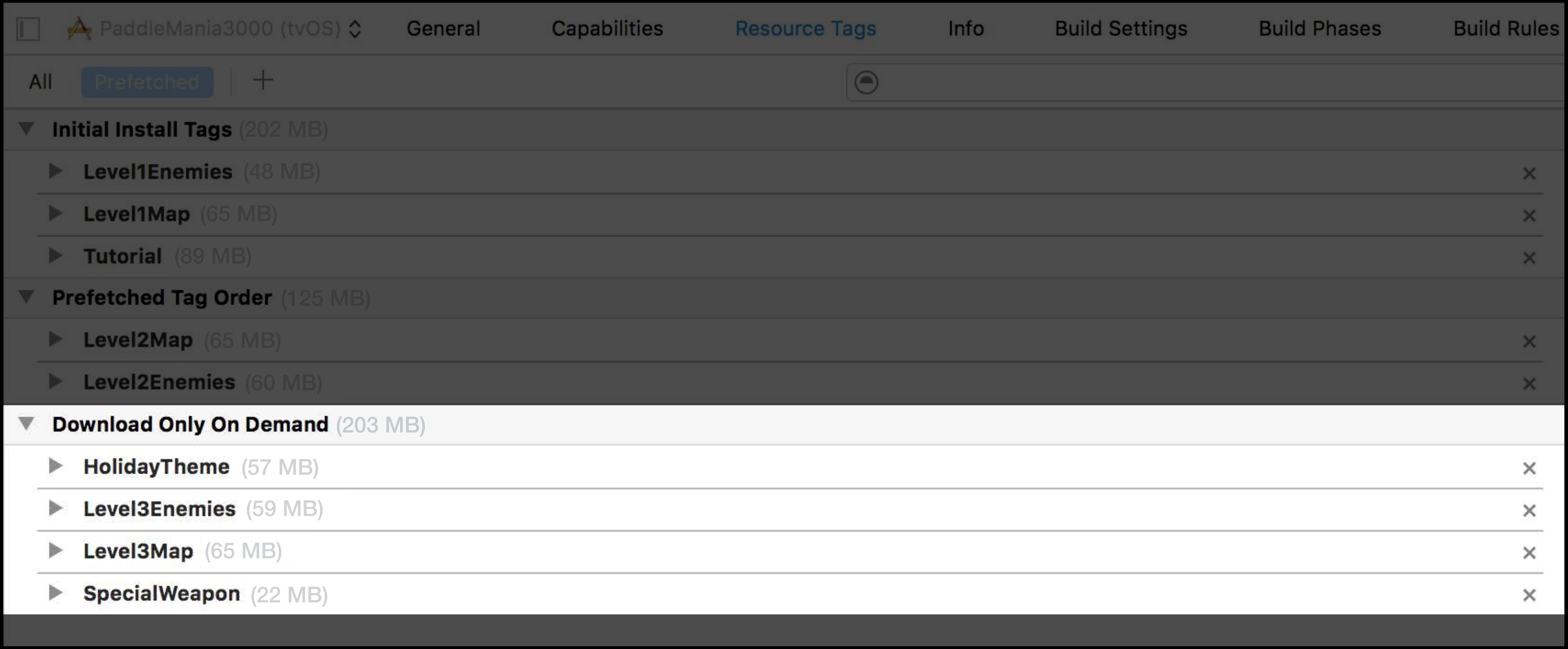
Initial Install Tags (202 MB)
- Level1Enemies (48 MB)
- Level1Map (65 MB)
- Tutorial (89 MB)

Prefetched Tag Order (125 MB)
- Level2Map (65 MB)
- Level2Enemies (60 MB)

Download Only On Demand (203 MB)
- HolidayTheme (57 MB)
- Level3Enemies (59 MB)
- Level3Map (65 MB)
- SpecialWeapon (22 MB)

Optimizing First Launch

Initial install and prefetched resources



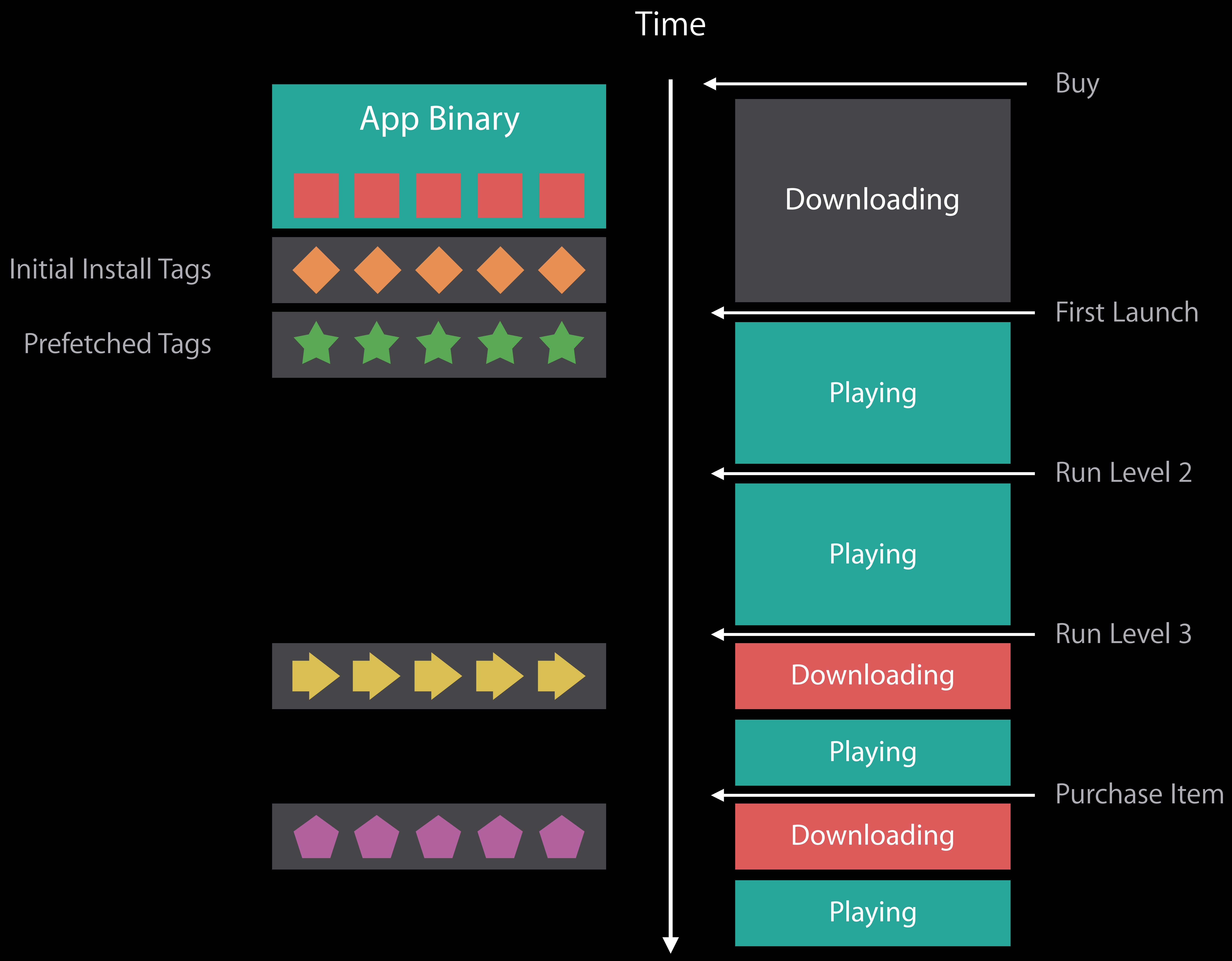
PaddleMania3000 (tvOS) | General | Capabilities | Resource Tags | Info | Build Settings | Build Phases | Build Rules

All | Prefetched | +

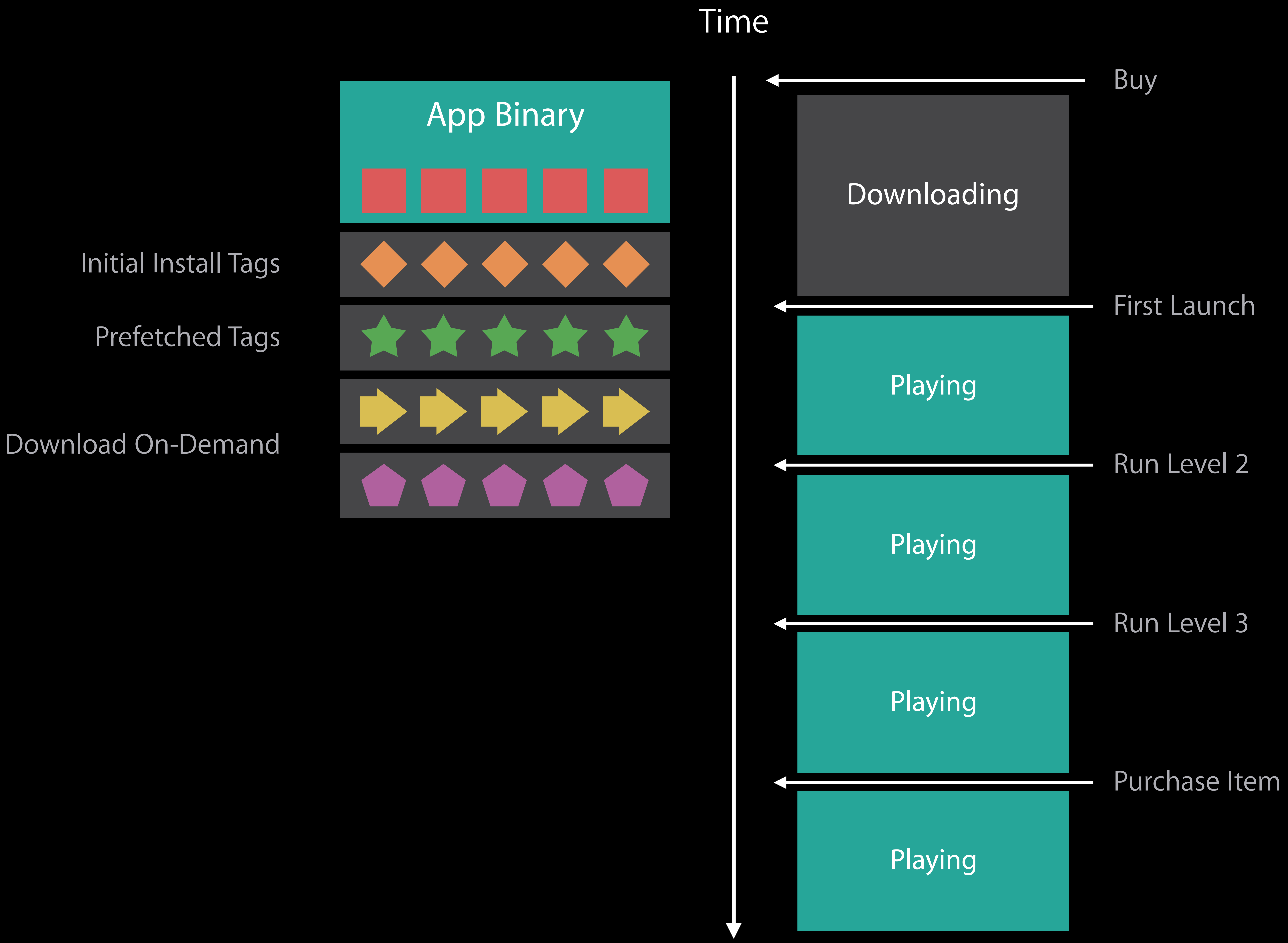
- **Initial Install Tags** (202 MB)
  - **Level1Enemies** (48 MB)
  - **Level1Map** (65 MB)
  - **Tutorial** (89 MB)
- **Prefetched Tag Order** (125 MB)
  - **Level2Map** (65 MB)
  - **Level2Enemies** (60 MB)
- **Download Only On Demand** (203 MB)
  - **HolidayTheme** (57 MB)
  - **Level3Enemies** (59 MB)
  - **Level3Map** (65 MB)
  - **SpecialWeapon** (22 MB)

Predictive Loading

Launch Timeline



Launch Timeline



Linear Access Pattern

Majority of assets will be used

Tag size isn't that critical

- Access tags early

Linear Access Pattern

Majority of assets will be used

Tag size isn't that critical

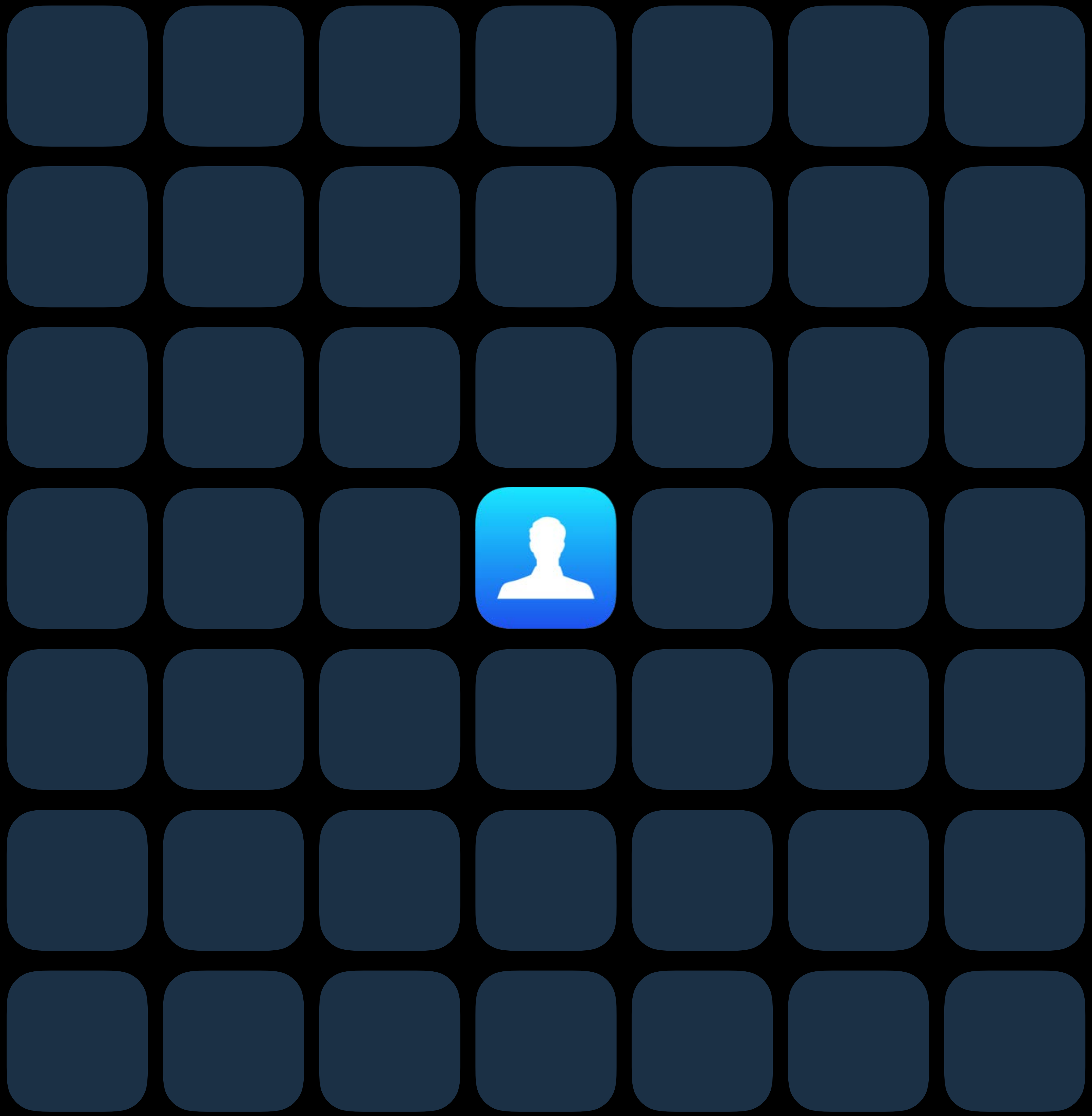
- Access tags early

Random Access Pattern

Access order is indeterminate

Use many tags

- Tag small groups of assets for progressive download and consumption
- Download sets of tags proactively
- End accessing does not mean deletion

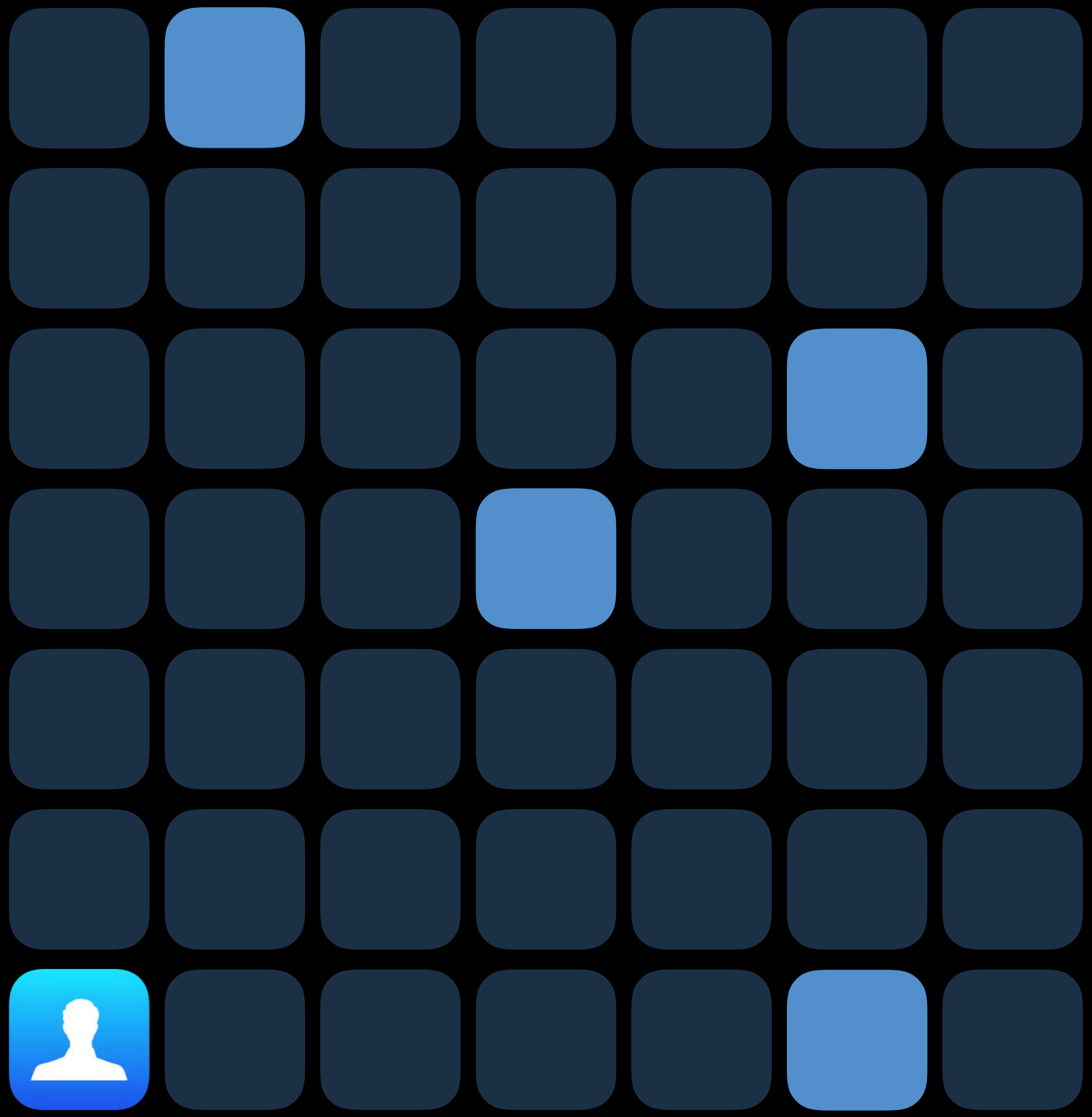


Random Access Pattern

Access order is indeterminate

Use many tags

- Tag small groups of assets for progressive download and consumption
- Download sets of tags proactively
- End accessing does not mean deletion



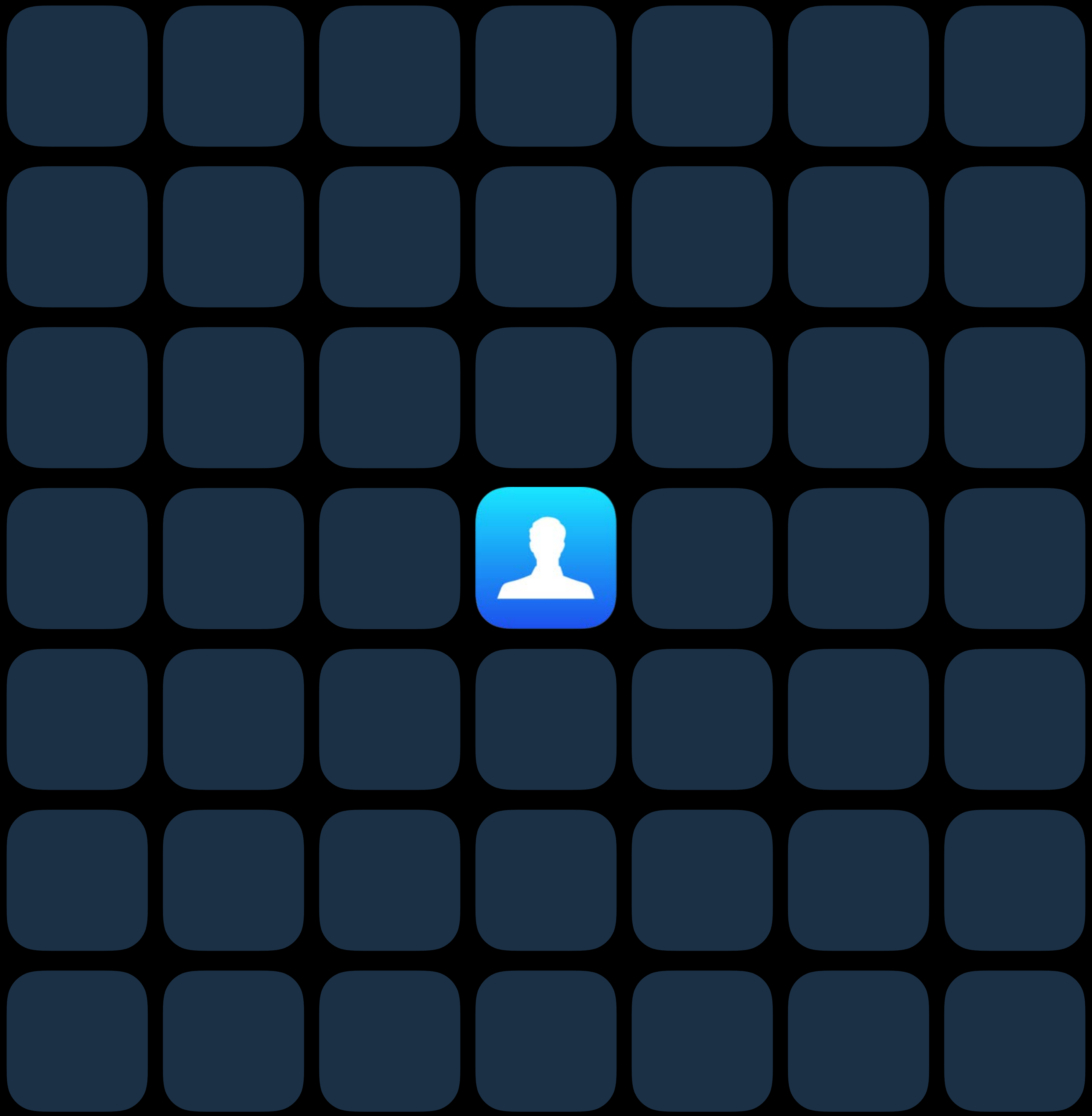
Explorative Access Pattern

Limited Prediction

- Many possibilities will not be used

Use many tags

- Load subset of possible resources
- Use hints to narrow the choices
- Quickly end accessing on unused resource requests



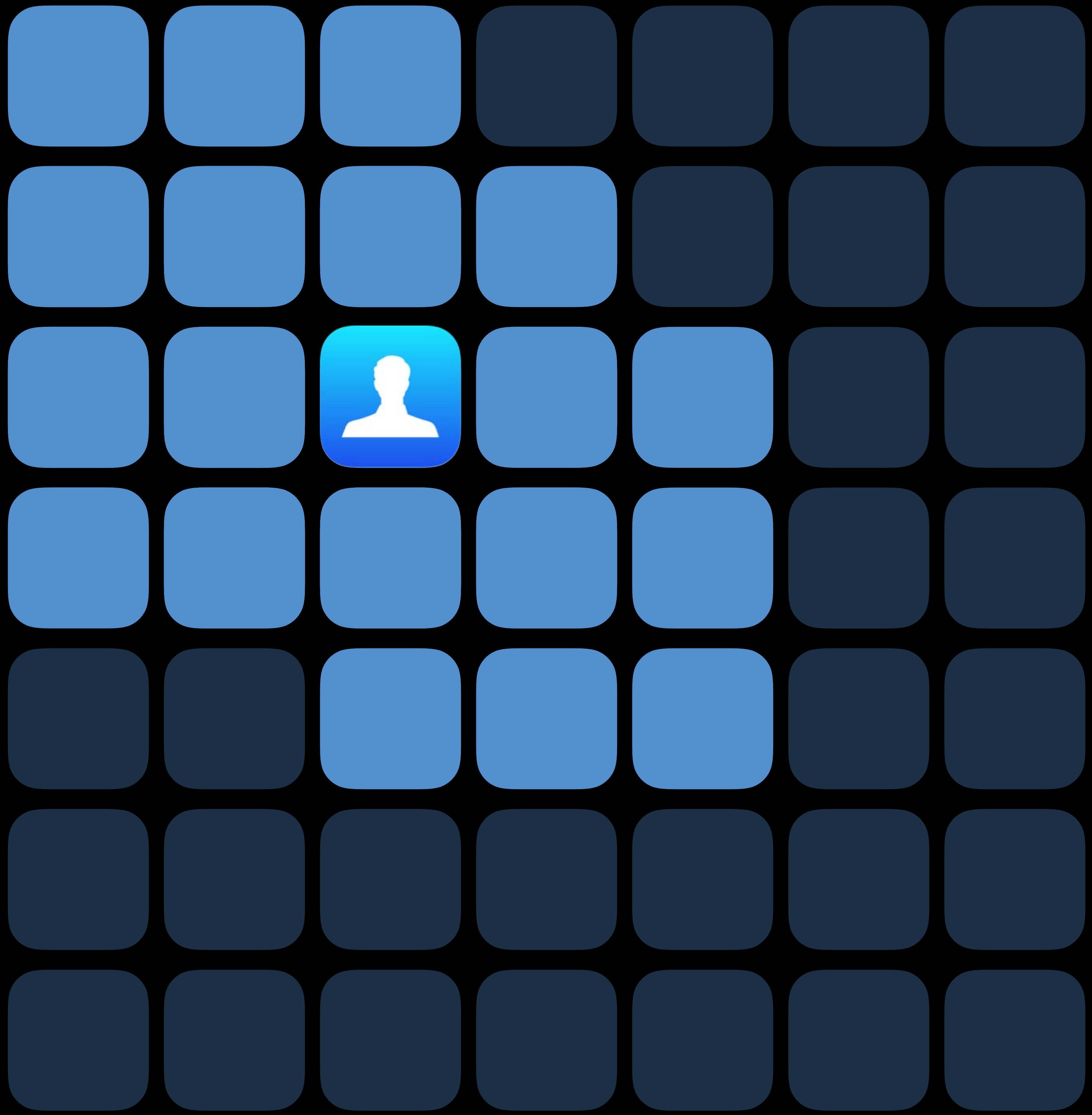
Explorative Access Pattern

Limited Prediction

- Many possibilities will not be used

Use many tags

- Load subset of possible resources
- Use hints to narrow the choices
- Quickly end accessing on unused resource requests



Implementation Details

On-Demand Resources

iOS App Bundle	Up to 4GB
tvOS App Bundle	Up to 200MB
On-Demand Resources	Up to 20GB
Initial Install Tags	Up to 2GB
Prefetch Tags	Up to 4GB – (Initial Install Tags)

On-Demand Resources

iOS App Bundle	Up to 4GB
tvOS App Bundle	Up to 200MB
On-Demand Resources	Up to 20GB
Initial Install Tags	Up to 2GB
Prefetch Tags	Up to 4GB – (Initial Install Tags)
Active Resources	Up to 2GB

On-Demand Resources

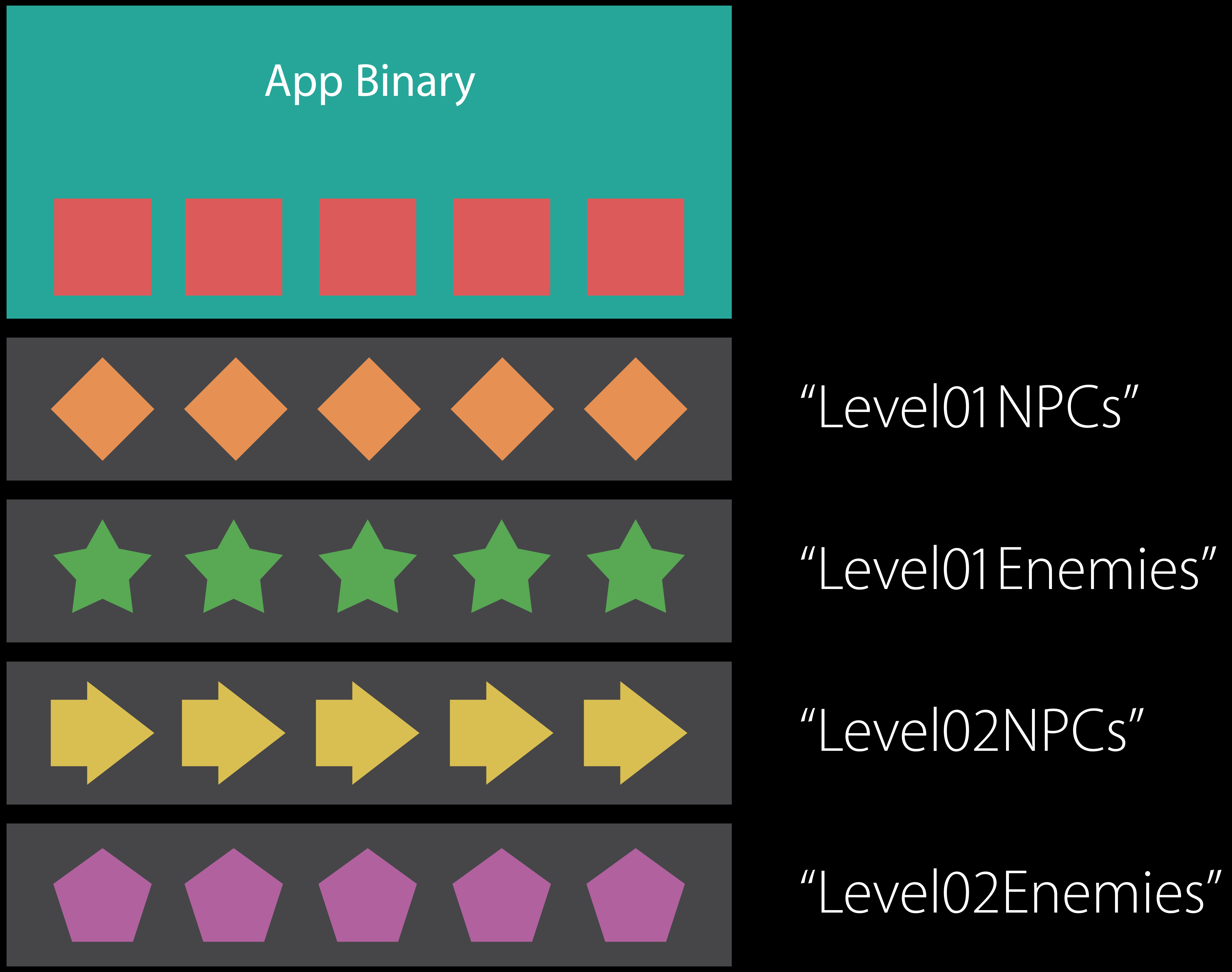
iOS App Bundle	Up to 4GB
tvOS App Bundle	Up to 200MB
On-Demand Resources	Up to 20GB
Initial Install Tags	Up to 2GB
Prefetch Tags	Up to 4GB – (Initial Install Tags)
Active Resources	Up to 2GB
One Tag	Up to 512MB (64MB recommended)

On-Demand Resources

iOS App Bundle	Up to 4GB
tvOS App Bundle	Up to 200MB
On-Demand Resources	Up to 20GB
Initial Install Tags	Up to 2GB
Prefetch Tags	Up to 4GB – (Initial Install Tags)
Active Resources	Up to 2GB
One Tag	Up to 512MB (64MB recommended)
Total Asset Packs	Up to 1000

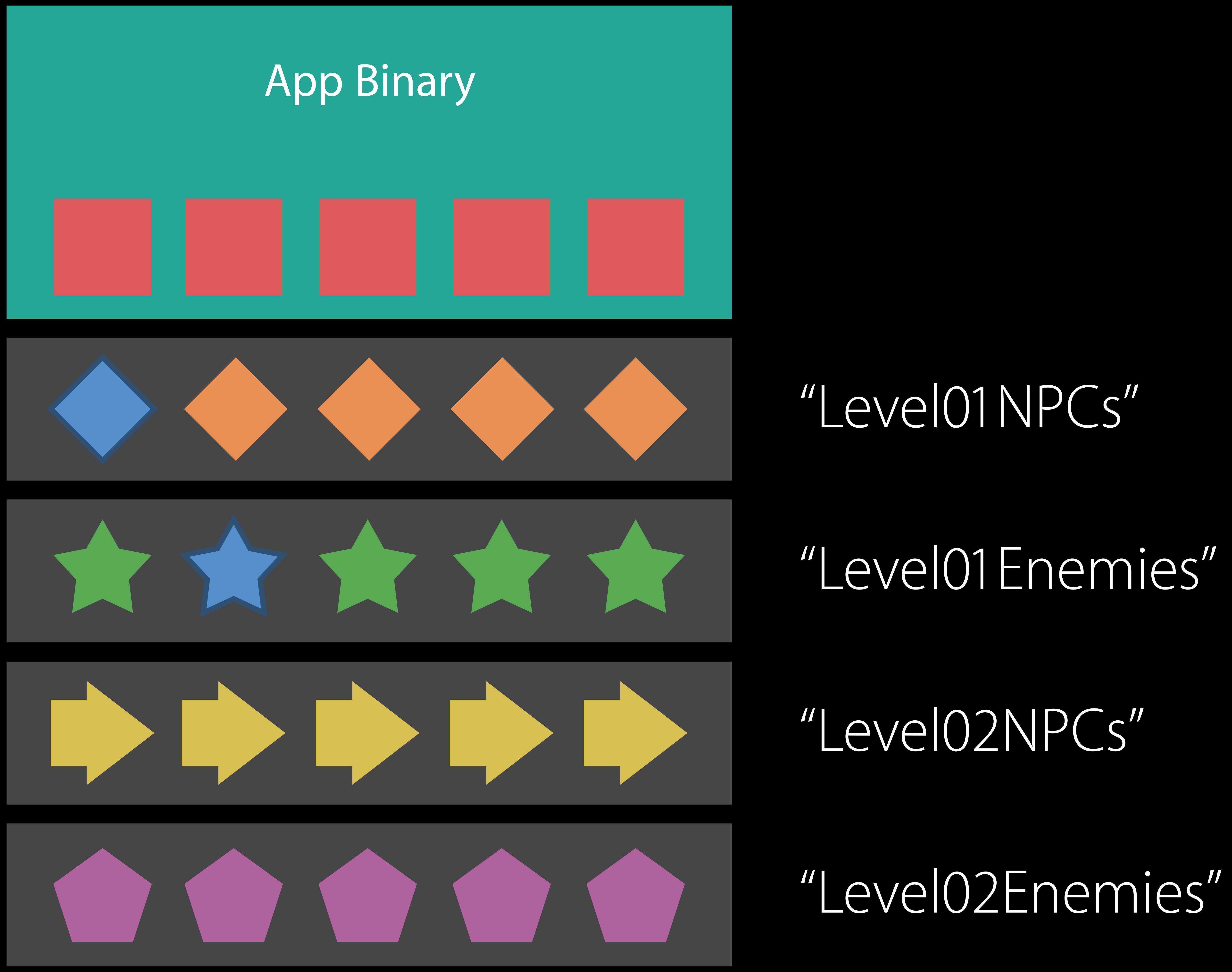
Asset Packs

GreatRPG.app



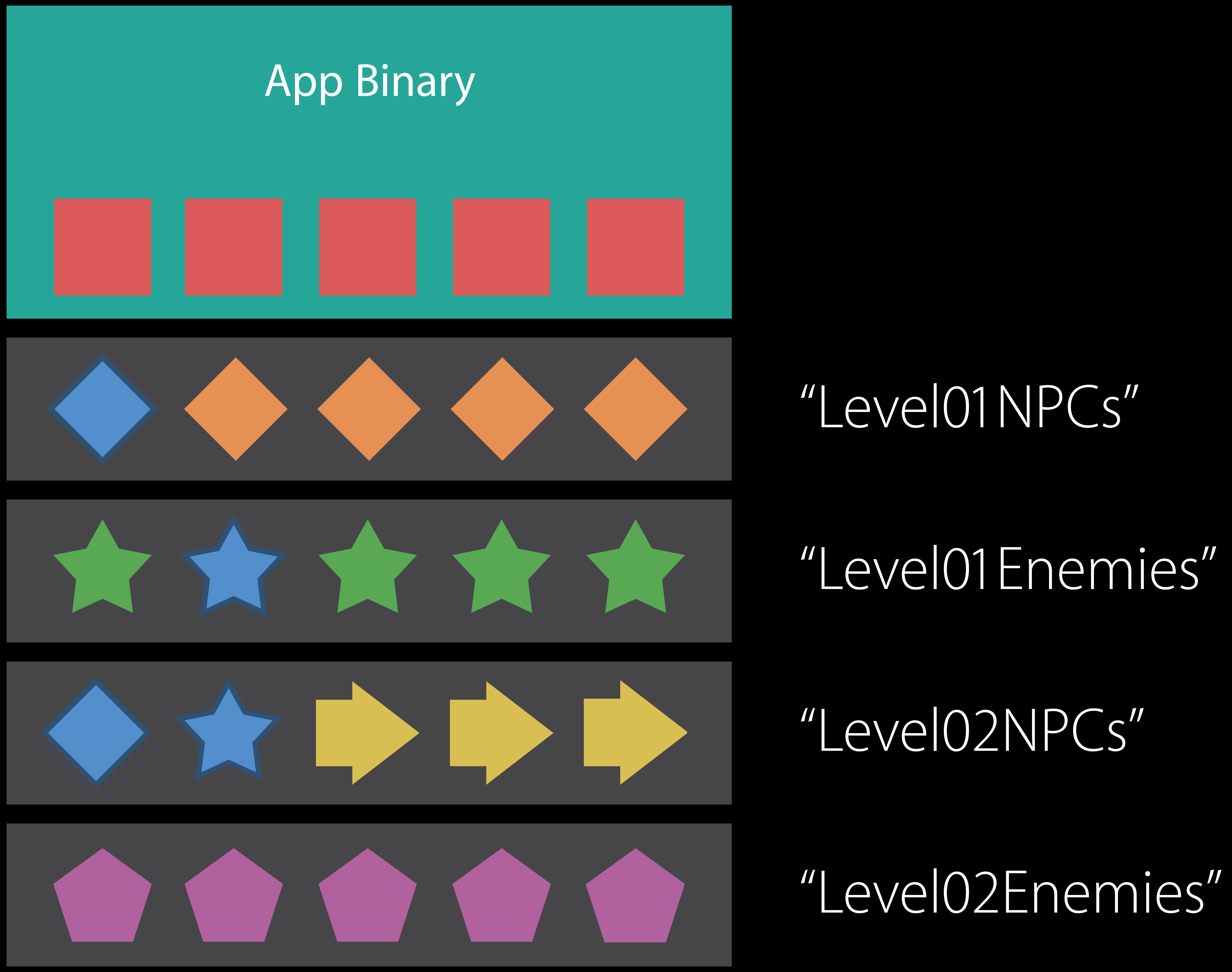
Asset Packs

GreatRPG.app



Asset Packs

GreatRPG.app



Asset Packs

GreatGame.app

🔶	"Level01NPCs"
★	"Level01Enemies"
➡	"Level02NPCs"
⬟	"Level02Enemies"
🔷	"Level01NPCs,""Level02NPCs"
★	"Level01Enemies,""Level02NPCs"

Asset Packs

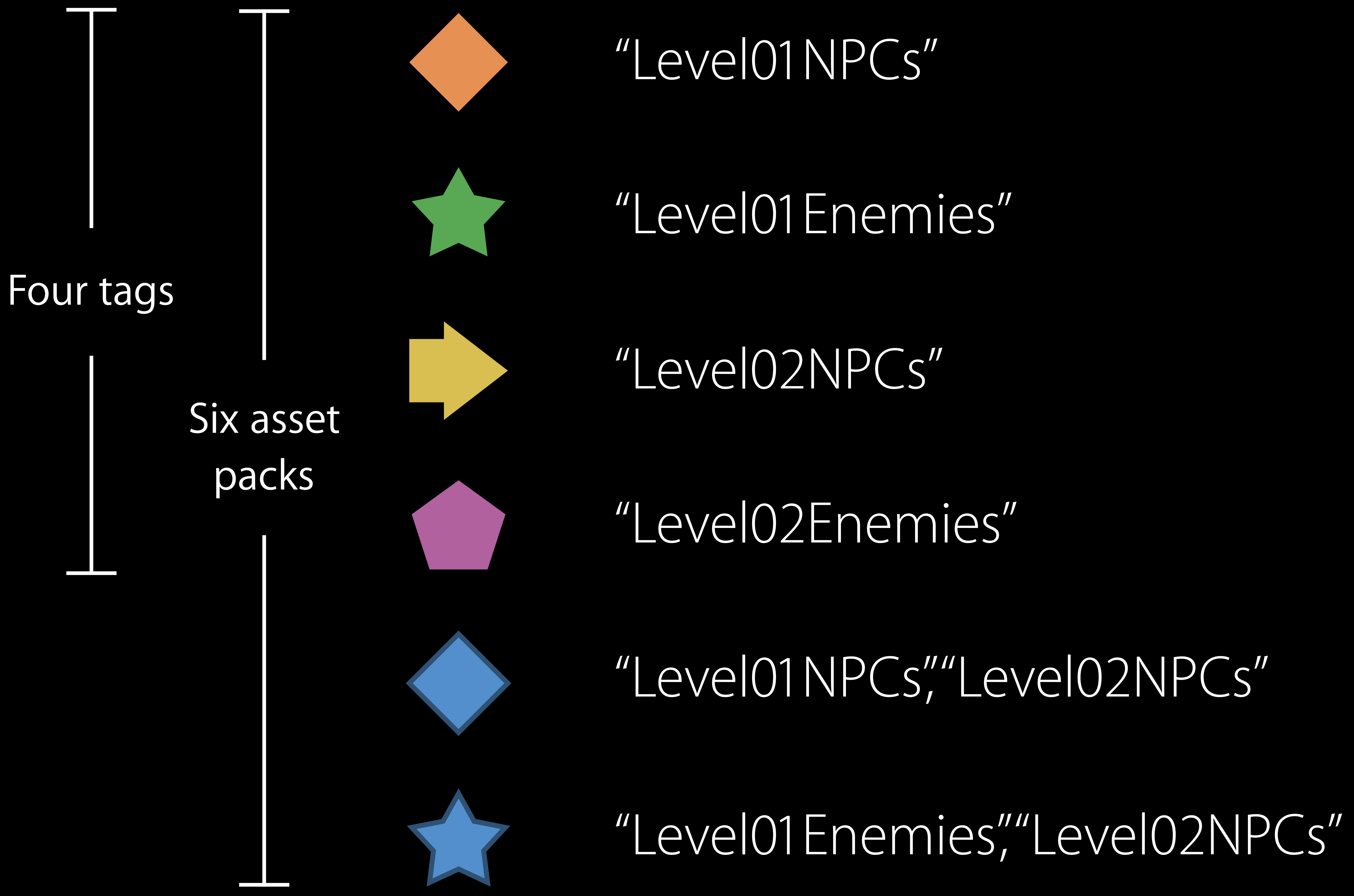
GreatGame.app

Four tags

- ◆ “Level01NPCs”
- ★ “Level01Enemies”
- ➡ “Level02NPCs”
- ⬟ “Level02Enemies”
- ◆ “Level01NPCs,”“Level02NPCs”
- ★ “Level01Enemies,”“Level02NPCs”

Asset Packs

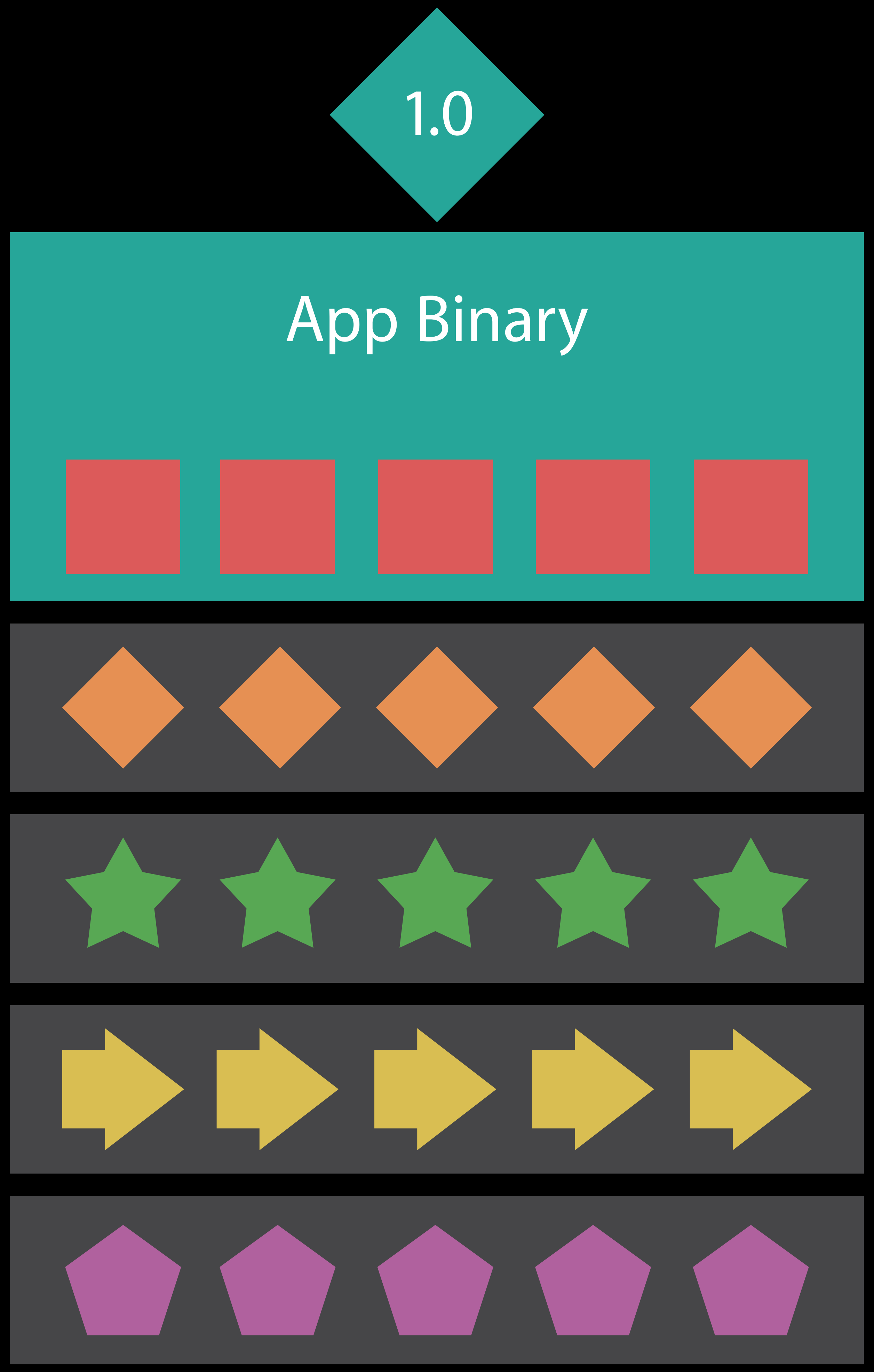
GreatGame.app



Optimizing ODR App Updates

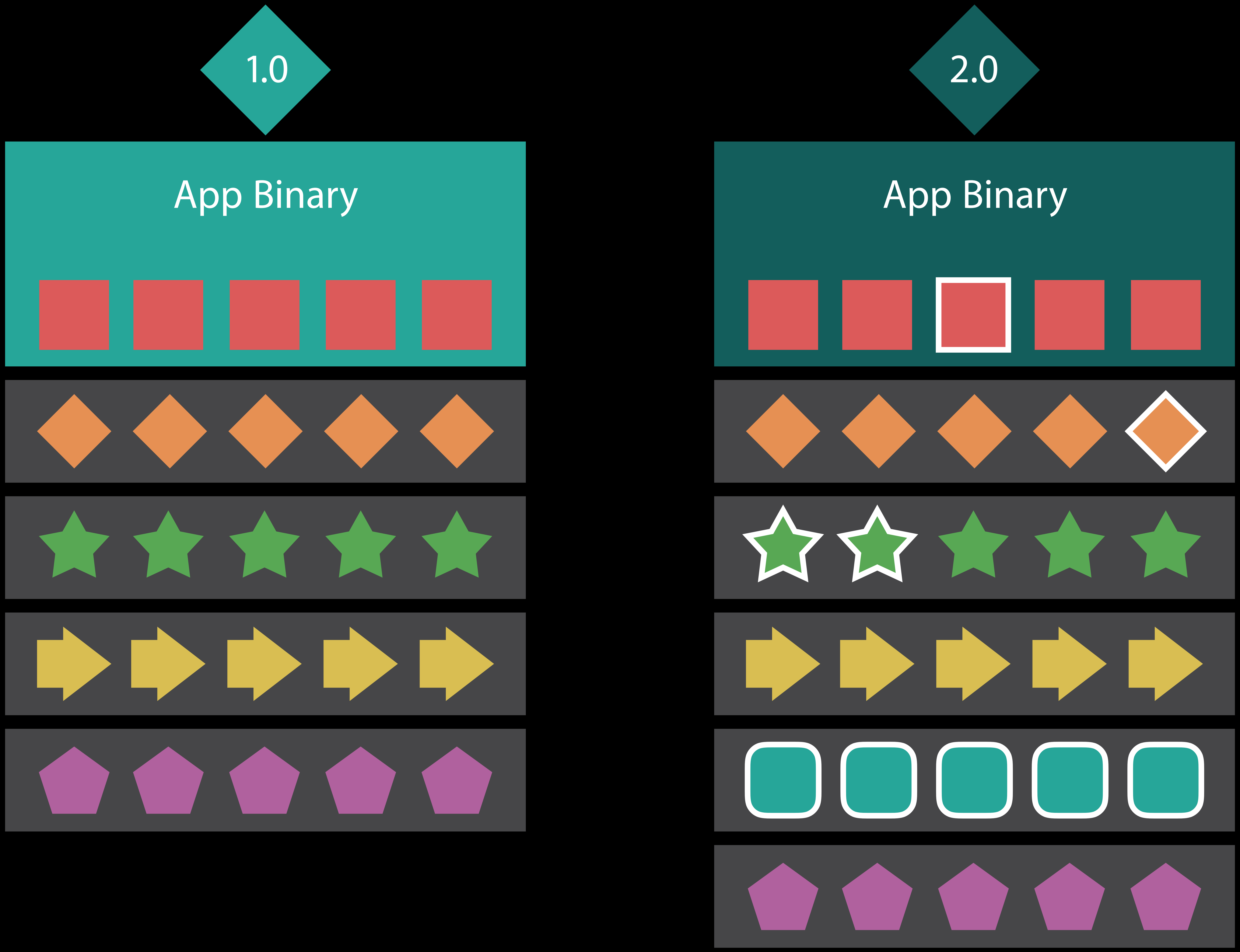
ODR App Updates

V1.0



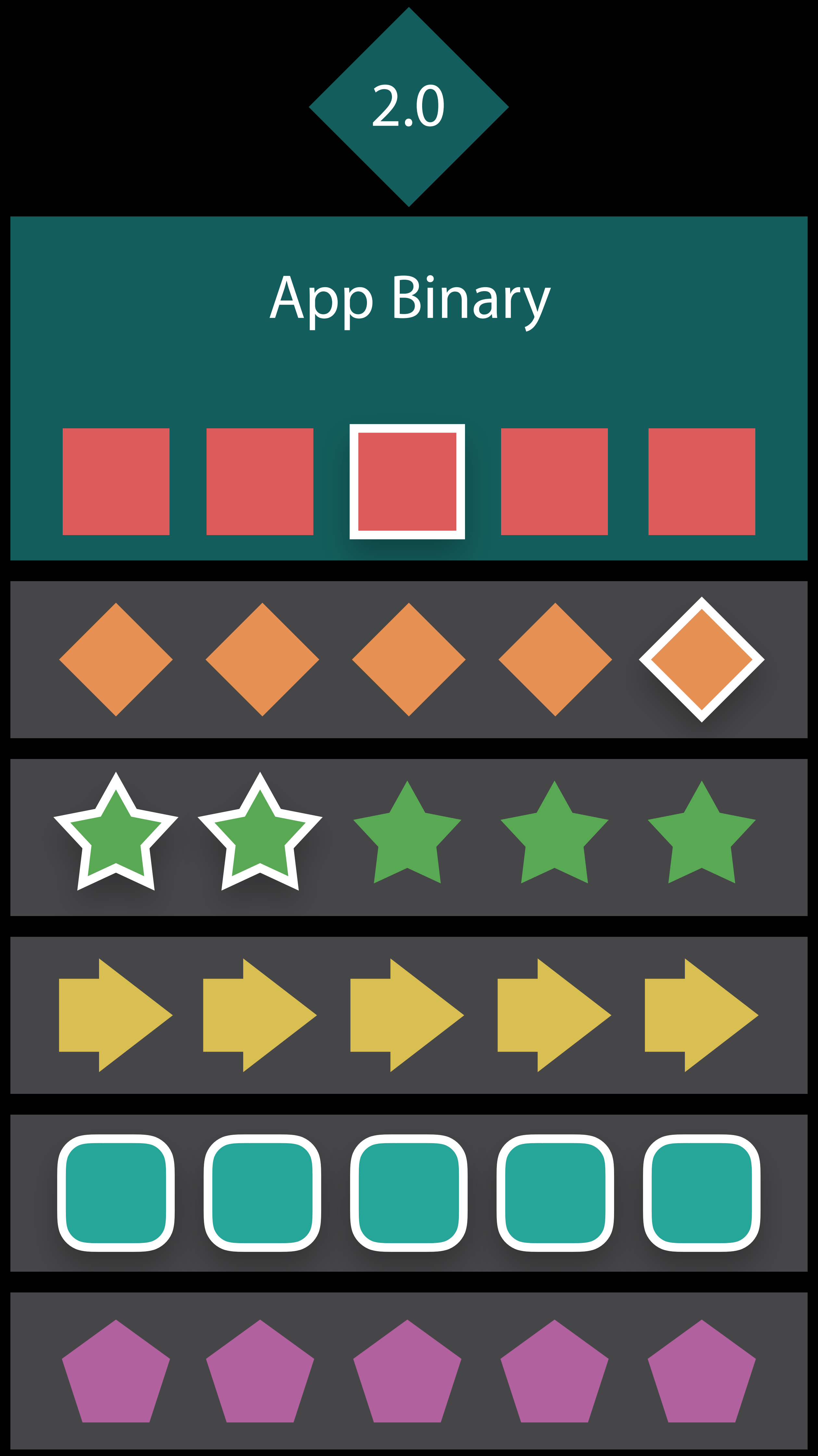
ODR App Updates

1.0 -> 2.0



ODR App Updates

Overview

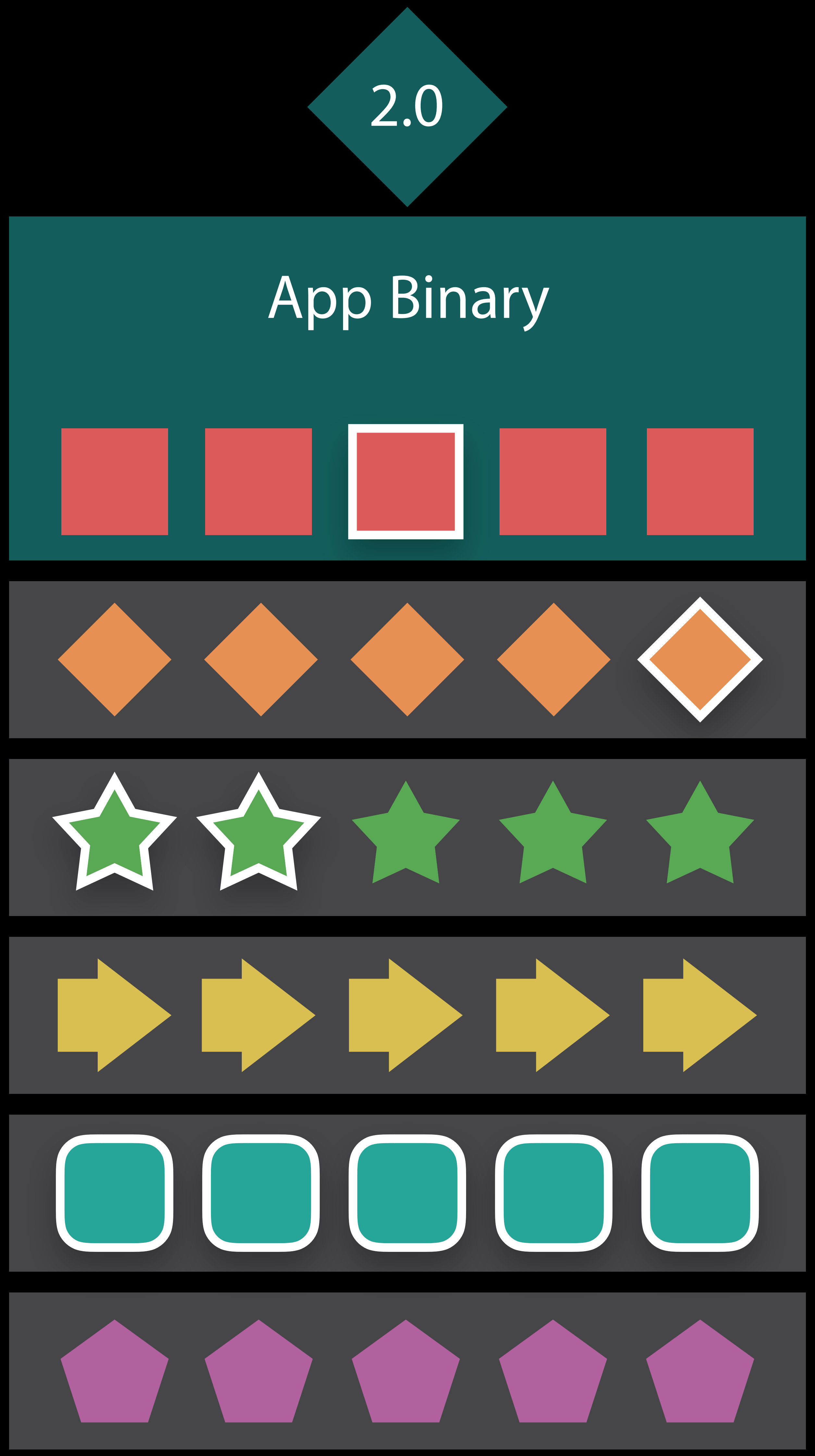


ODR App Updates

Overview

Updated resources

- Redownload only when accessed



ODR App Updates

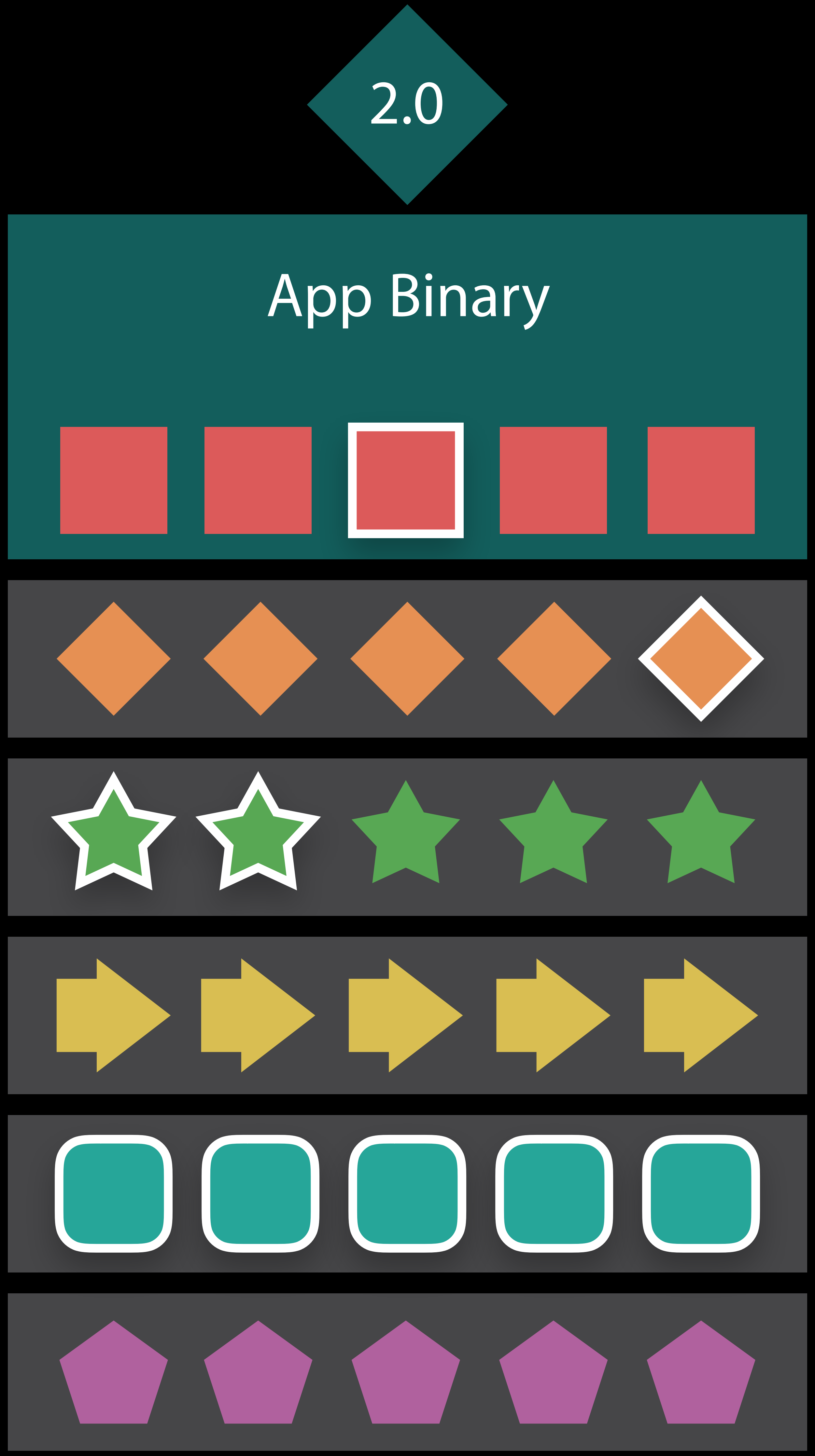
Overview

Updated resources

- Redownload only when accessed

Unchanged resources

- Usually remain on device
- Can be accessed without redownload



ODR App Updates

Overview

Updated resources

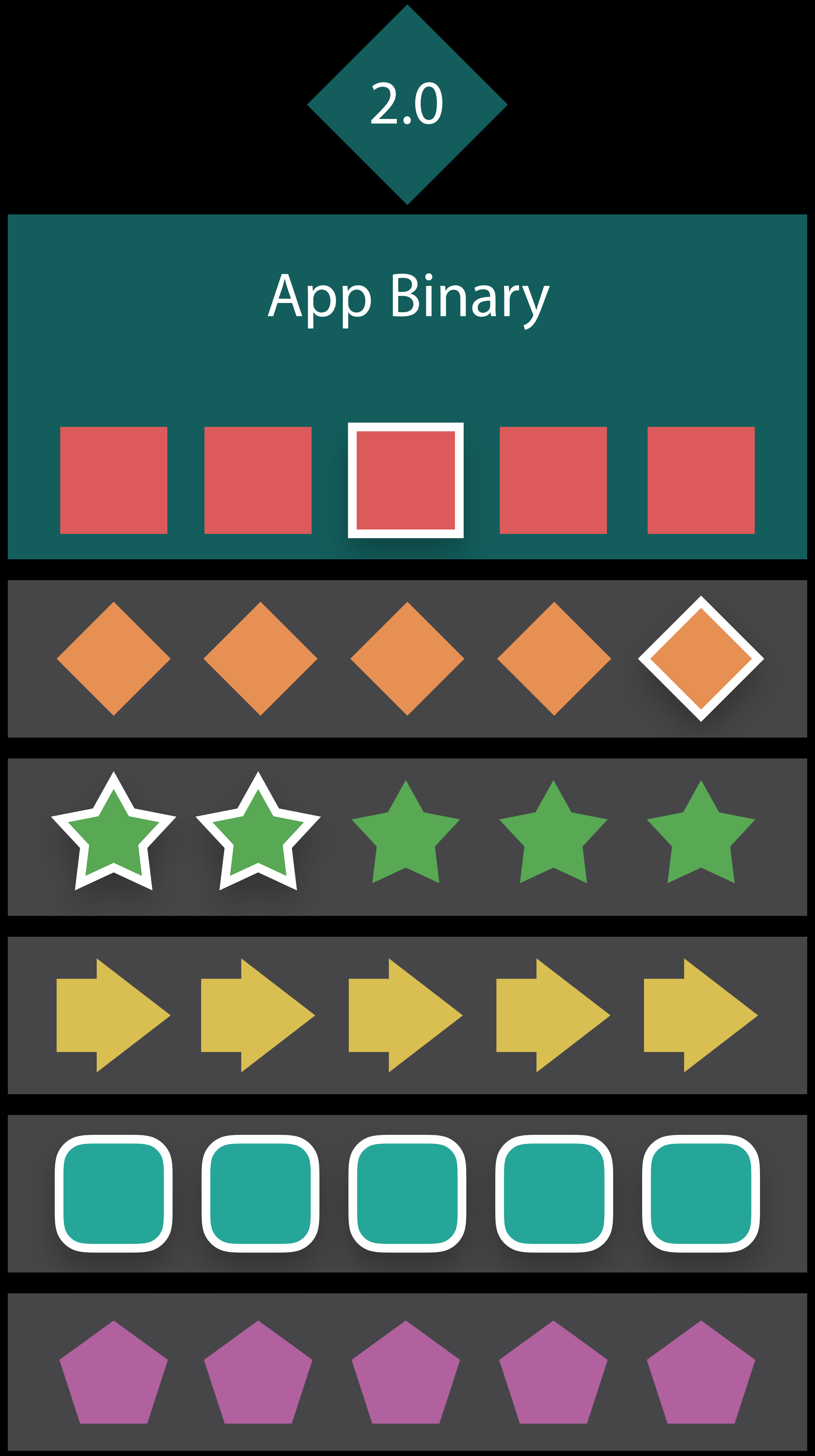
- Redownload only when accessed

Unchanged resources

- Usually remain on device
- Can be accessed without redownload

New resources

- Download only when accessed



ODR App Updates

Best practices

Avoid making unnecessary modifications to tagged resources

- One changed resource triggers redownload of the asset pack
- Consider addendum tags;“Level 01” and “Level 01 Update 1”

ODR App Updates

Best practices

Avoid making unnecessary modifications to tagged resources

- One changed resource triggers redownload of the asset pack
- Consider addendum tags; "Level 01" and "Level 01 Update 1"

Keep tags consistent from version to version

ODR App Updates

Best practices

Avoid making unnecessary modifications to tagged resources

- One changed resource triggers redownload of the asset pack
- Consider addendum tags; "Level 01" and "Level 01 Update 1"

Keep tags consistent from version to version

Design with separation of updatable content vs. static content

Intelligent Content Caching

How it works

Intelligent Content Caching

How it works

Resources may be purged when the system demands disk space

- Ending access does not mean deletion

Intelligent Content Caching

How it works

Resources may be purged when the system demands disk space

- Ending access does not mean deletion

Variables that inform purge order

- Last used timestamp
- Preservation priority
  - Isolated to your application
- Application-running state

Intelligent Content Caching

How it works

Resources may be purged when the system demands disk space

- Ending access does not mean deletion

Variables that inform purge order

- Last used timestamp
- Preservation priority
  - Isolated to your application
- Application-running state

Don’t use tmp or caches

- Purged first. Purged completely.

Conclusion

Use On-Demand Resources

On-Demand Resources

Smaller main app bundle

- Faster initial download

On-Demand Resources

Smaller main app bundle

- Faster initial download

Richer app content

- Up to 20GB available on-demand

On-Demand Resources

Smaller main app bundle

- Faster initial download

Richer app content

- Up to 20GB available on-demand

More apps installed and ready to run

- Reduces need to manage storage

More Information

https://developer.apple.com/wwdc16/221

Related Sessions

What’s New in tvOS	Presidio	Tuesday 3:00PM
Designing for tvOS	Presidio	Tuesday 4:00PM
Controlling Game Input for Apple TV	Mission	Wednesday 5:00PM

Lab

On-Demand Resources Lab	Frameworks Lab B	Thursday 11:00AM

WWDC16