

Robert Ricci

Workshop on Future of Resource Representations in GENI

March 18, 2010

The ProtoGENI Vision for GENI Resource Representation



**Resources in GENI
are complicated**

**Representations of them
shouldn't be**

**Lots of different software
packages have to process
resource descriptions**

Keep It Simple

Keep It Simple
(and **Extensible**) and

Keep It Simple
(and **Extensible**) and
Stupid

KISSES

What I'll Talk About

- (Few) details of the ProtoGENI RSpec
- Lifecycle of the RSpec
- Identifiers
- Extensions and why they are critical
- Interoperability concerns

Where We Are Now

- **Data structure, not a language**
- Working prototype RSpec (v2 on the way)
- Expressed as an XML schema (using Relax-NG)
 - <http://www.protogeni.net/trac/protogeni/wiki/RSpec>
- Key objects nodes, interfaces, links
 - Describe “lowest level” at which experimenter operates
- Used to create slivers:
 - Raw PCs, VMs, VLANs, tunnels
- Slice Embedding Service that understands it
- External references and stitching
- Extensions using namespaces

Types of RSpecs

- Advertisement: "Catalog"
- Request: "Purchase Order"
- Ticket: "Receipt"
- Manifest: "Packing Slip"

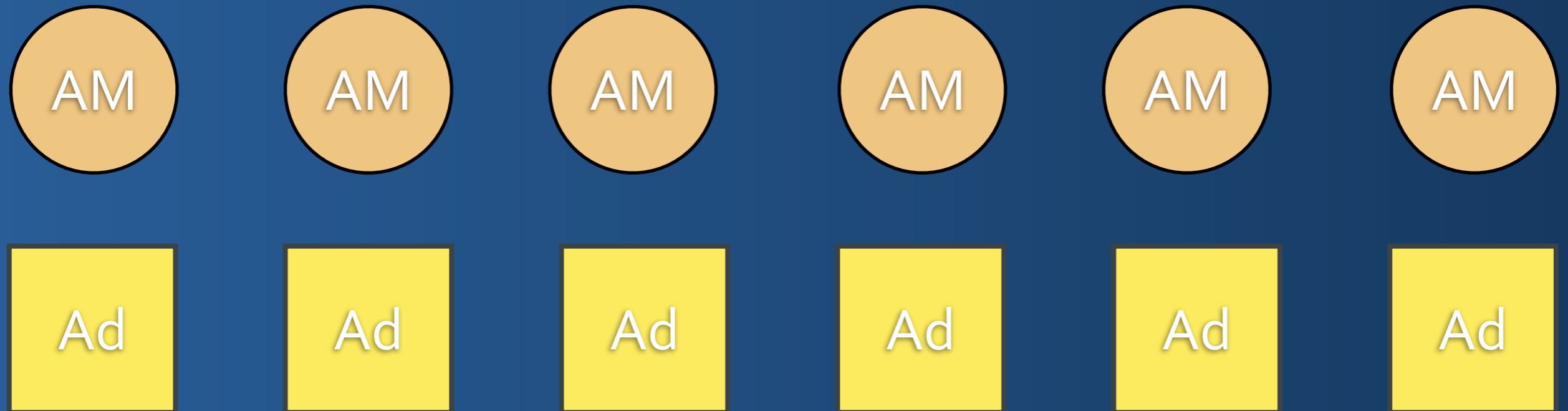
RSpec Lifecycle

- **Progressive annotation**
- User creates **request** (bound or unbound)
- Passes to a Slice Embedding Service
 - Annotates with physical resources selected
 - Maybe more than one
- Gives to CM
 - CM signs (**ticket**)
- **Manifest** returned by CM

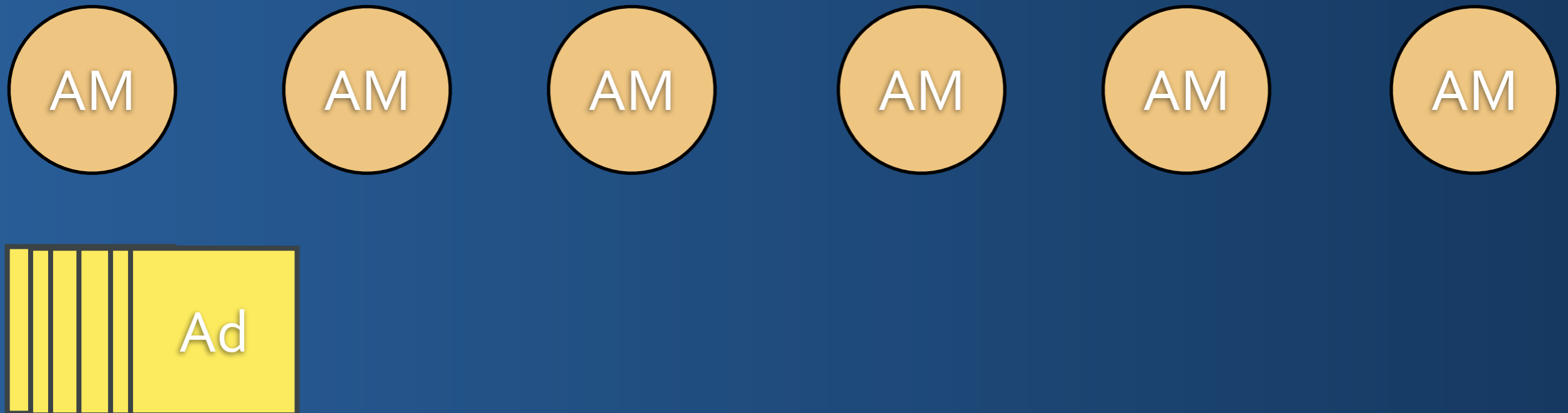
RSpec Lifecycle



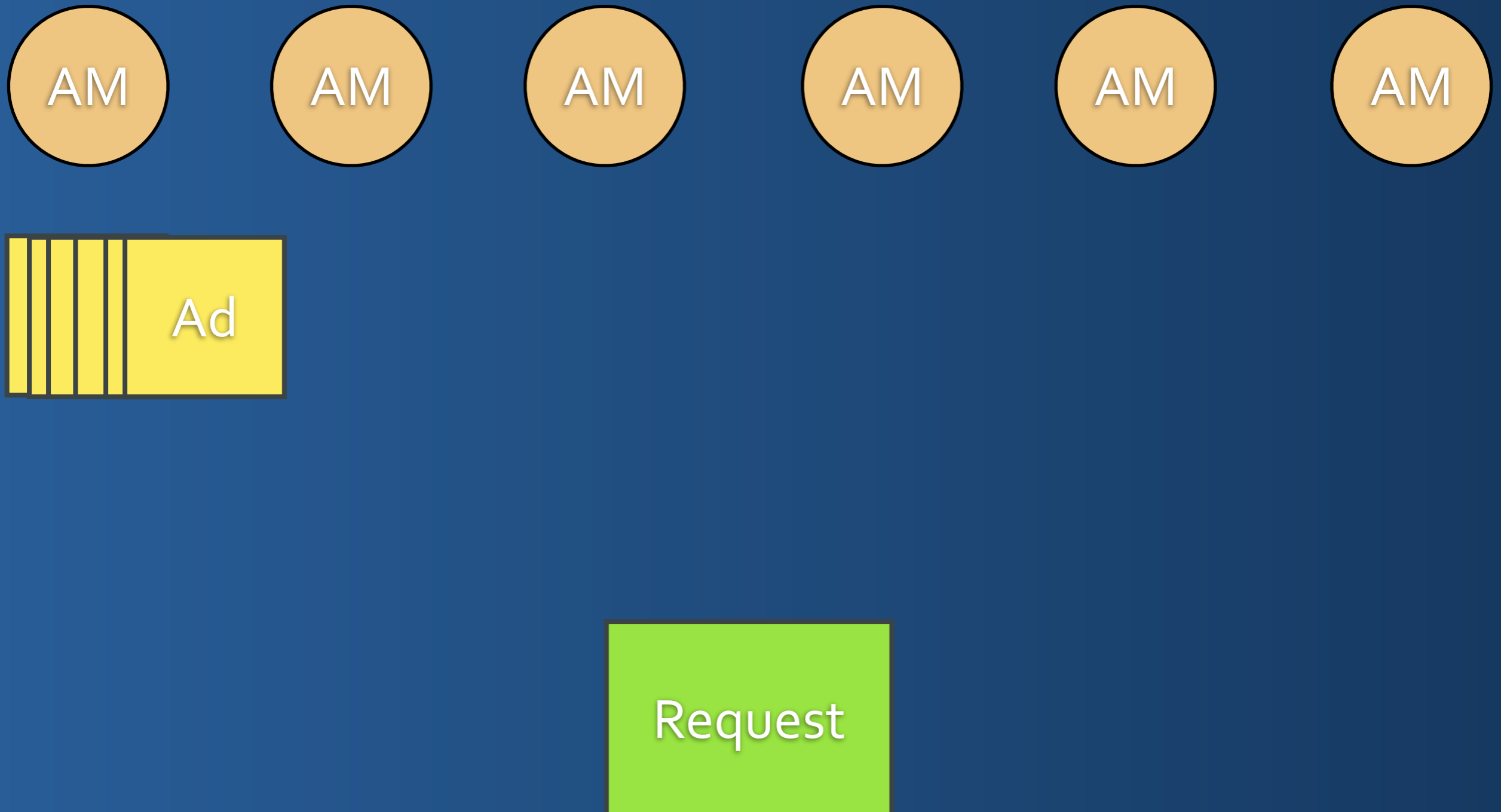
RSpec Lifecycle



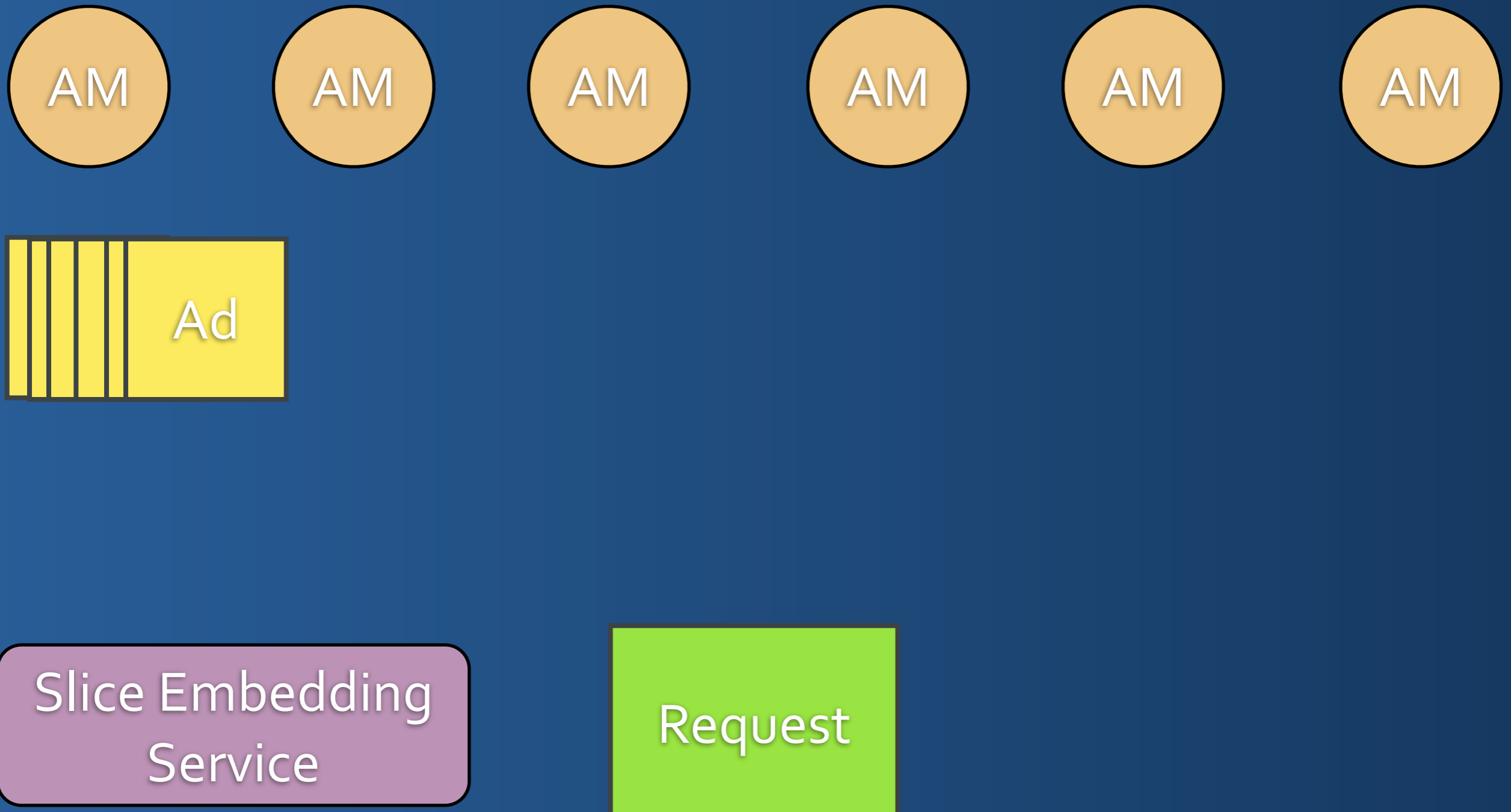
RSpec Lifecycle



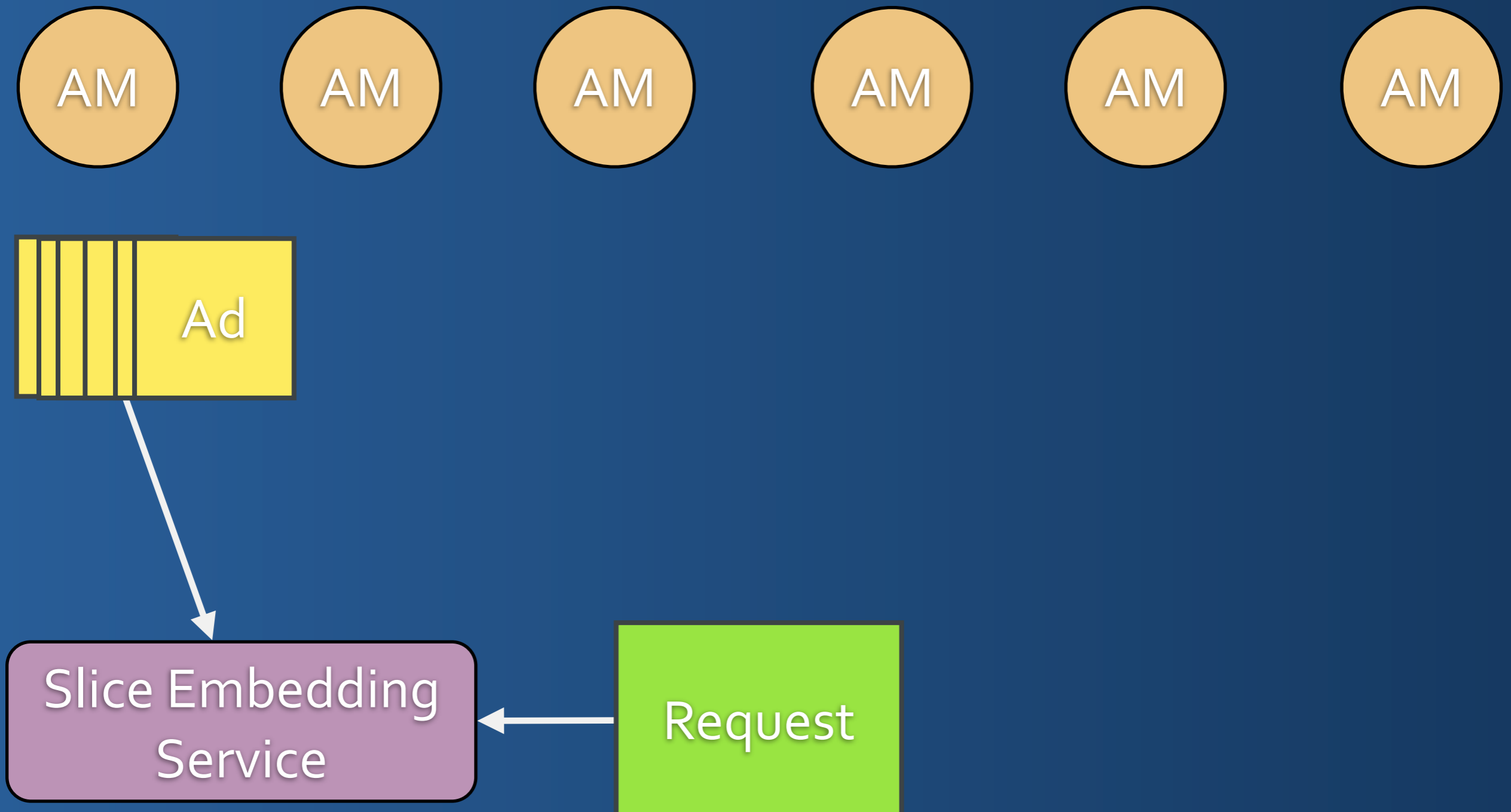
RSpec Lifecycle



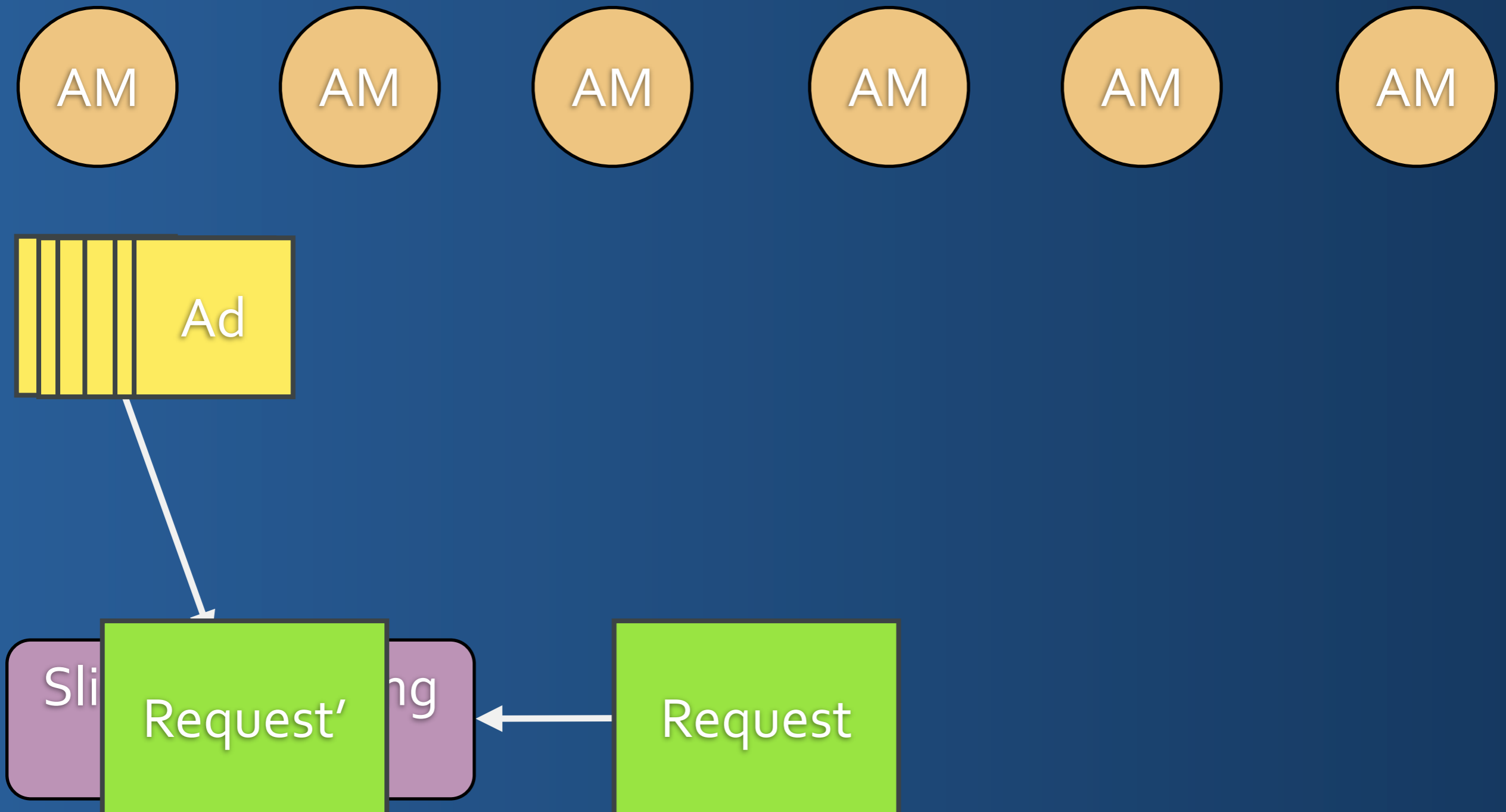
RSpec Lifecycle



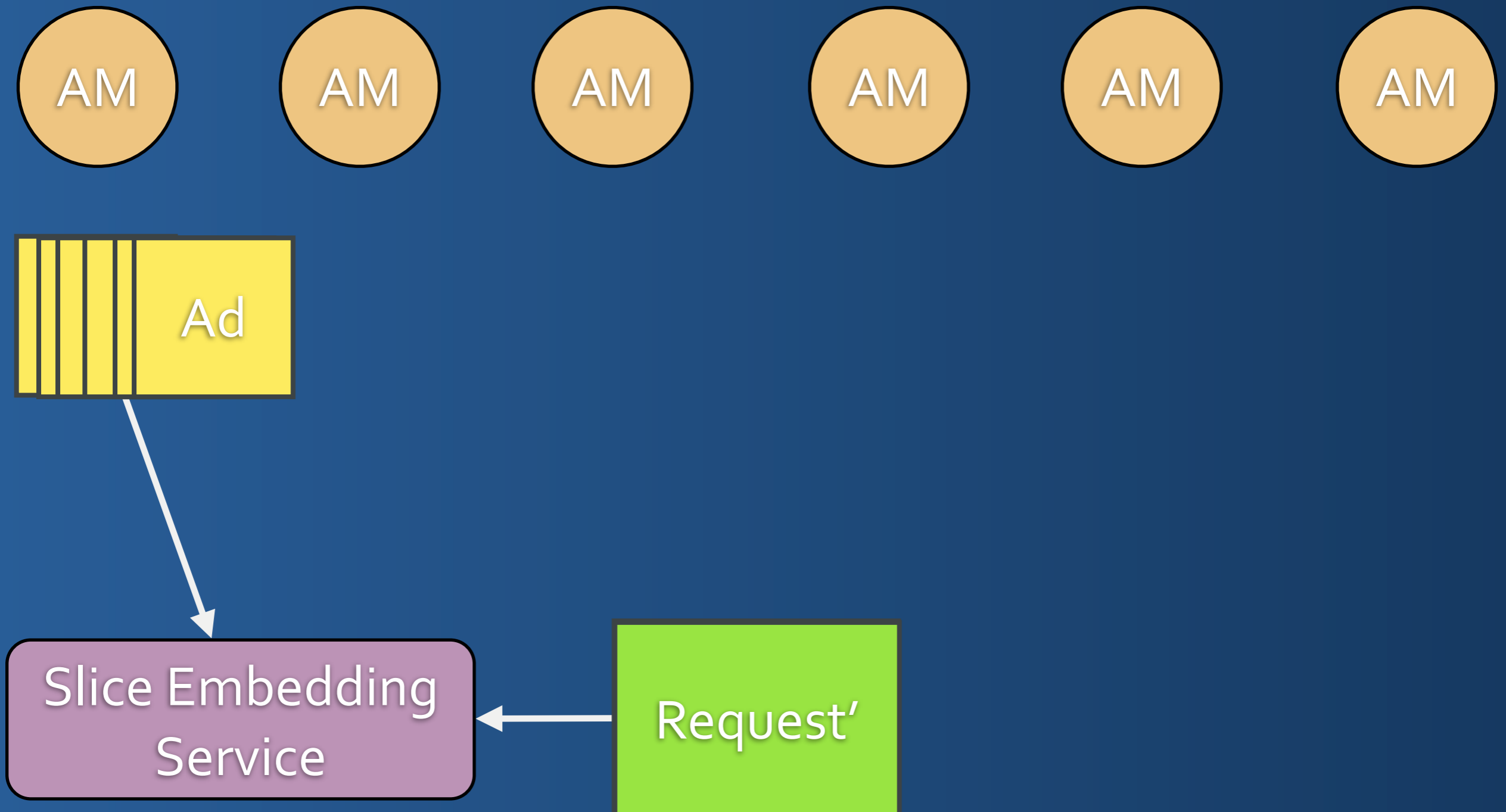
RSpec Lifecycle



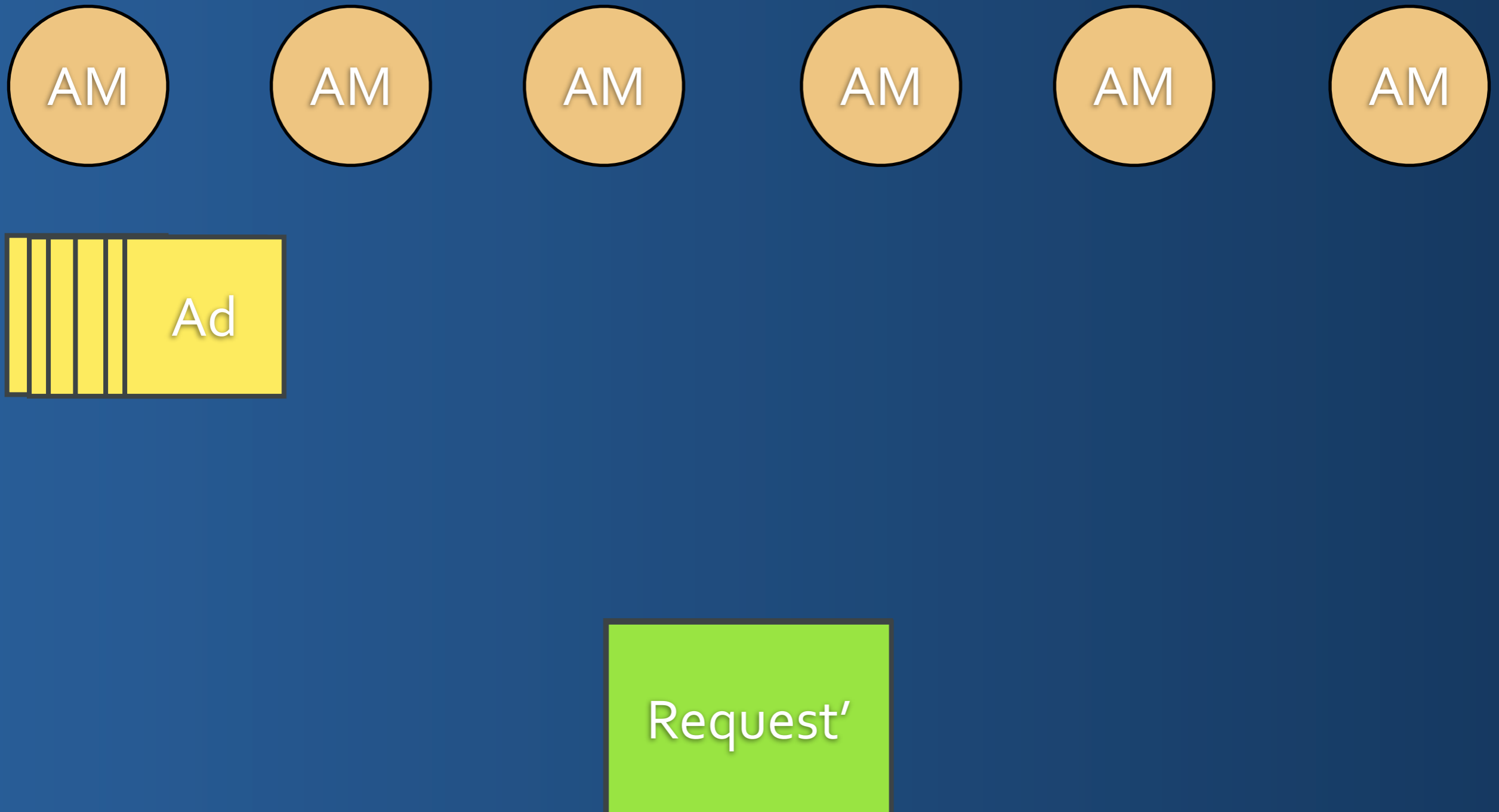
RSpec Lifecycle



RSpec Lifecycle



RSpec Lifecycle



RSpec Lifecycle

AM

AM

AM

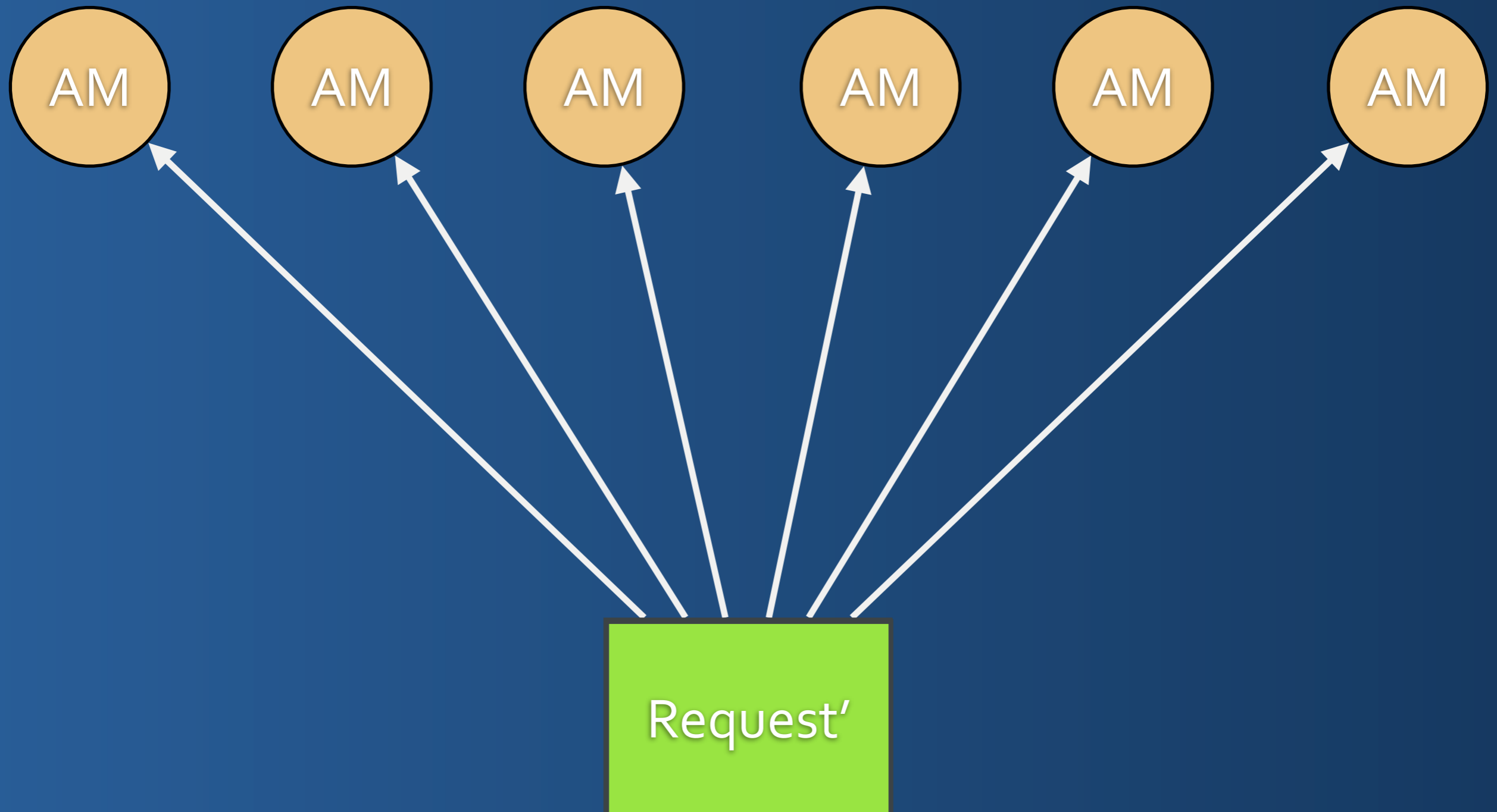
AM

AM

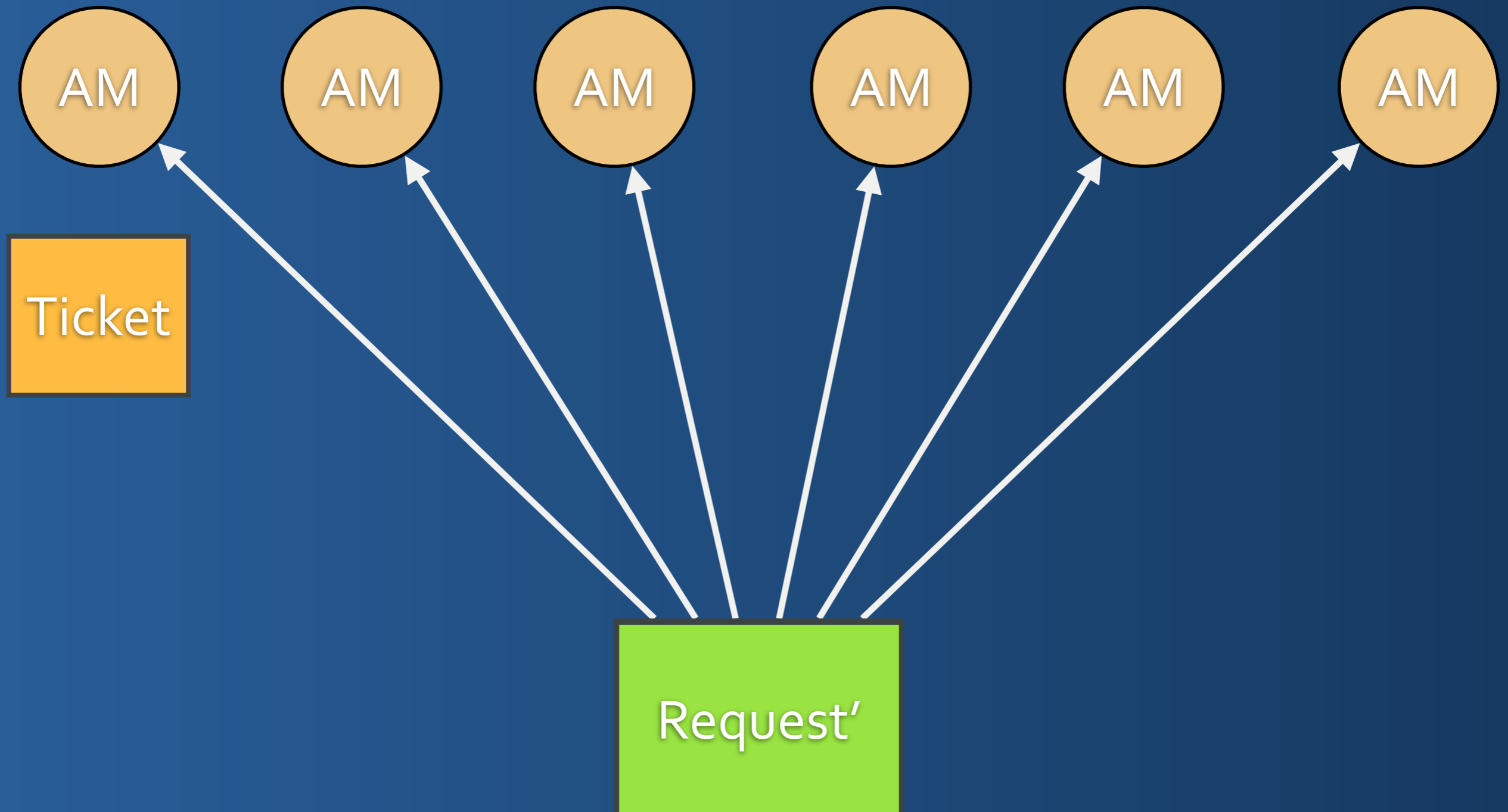
AM

Request'

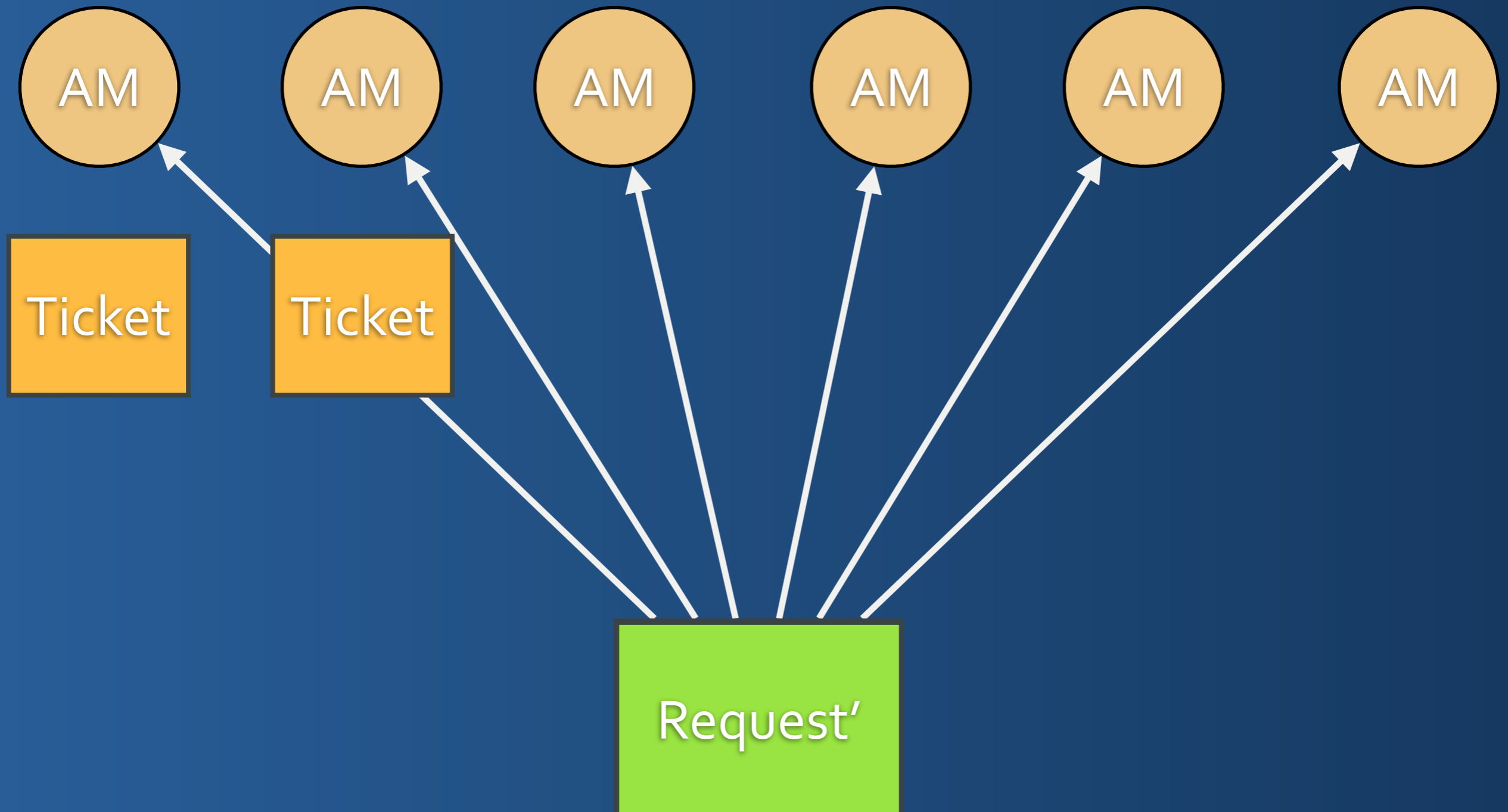
RSpec Lifecycle



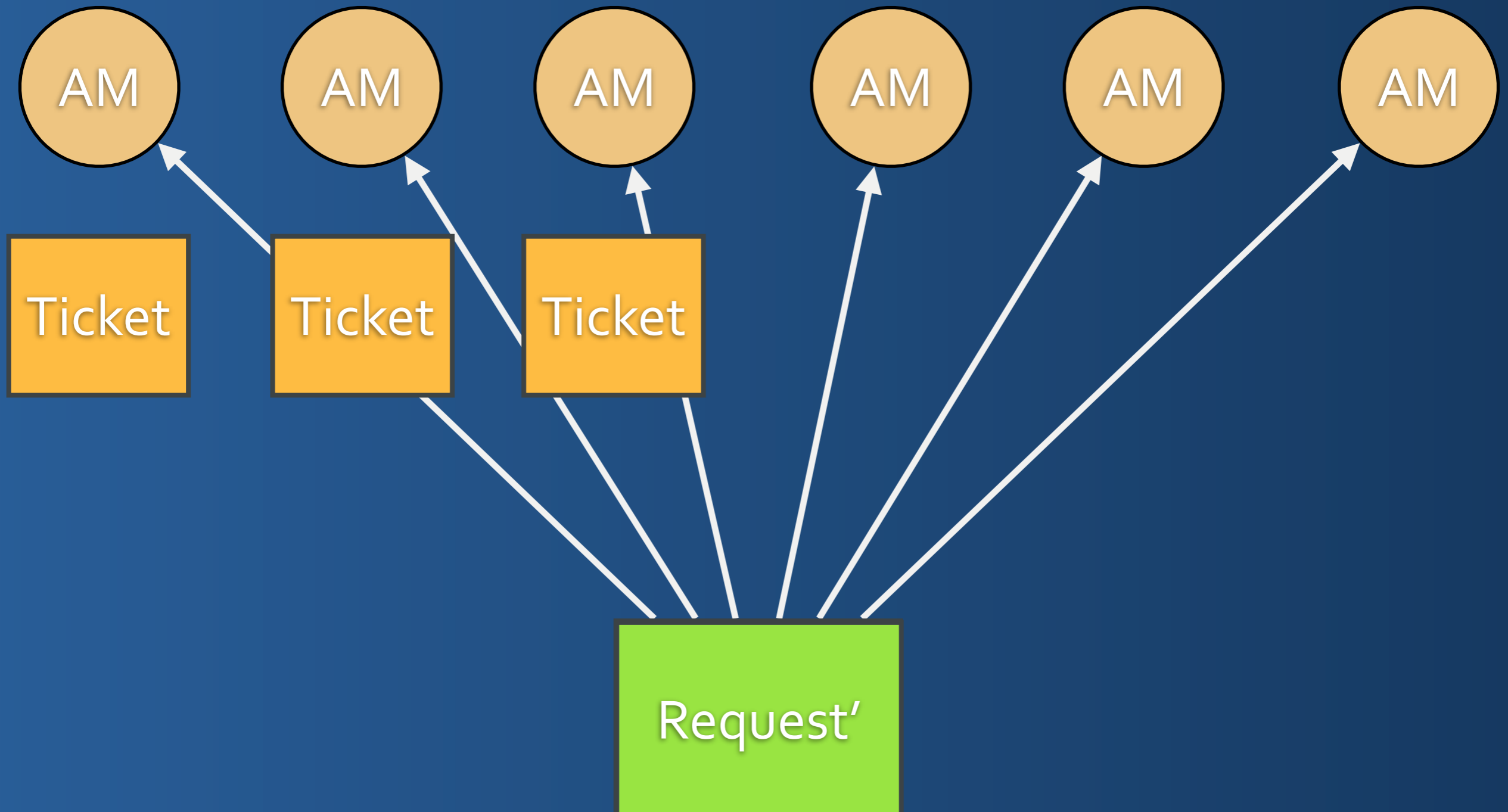
RSpec Lifecycle



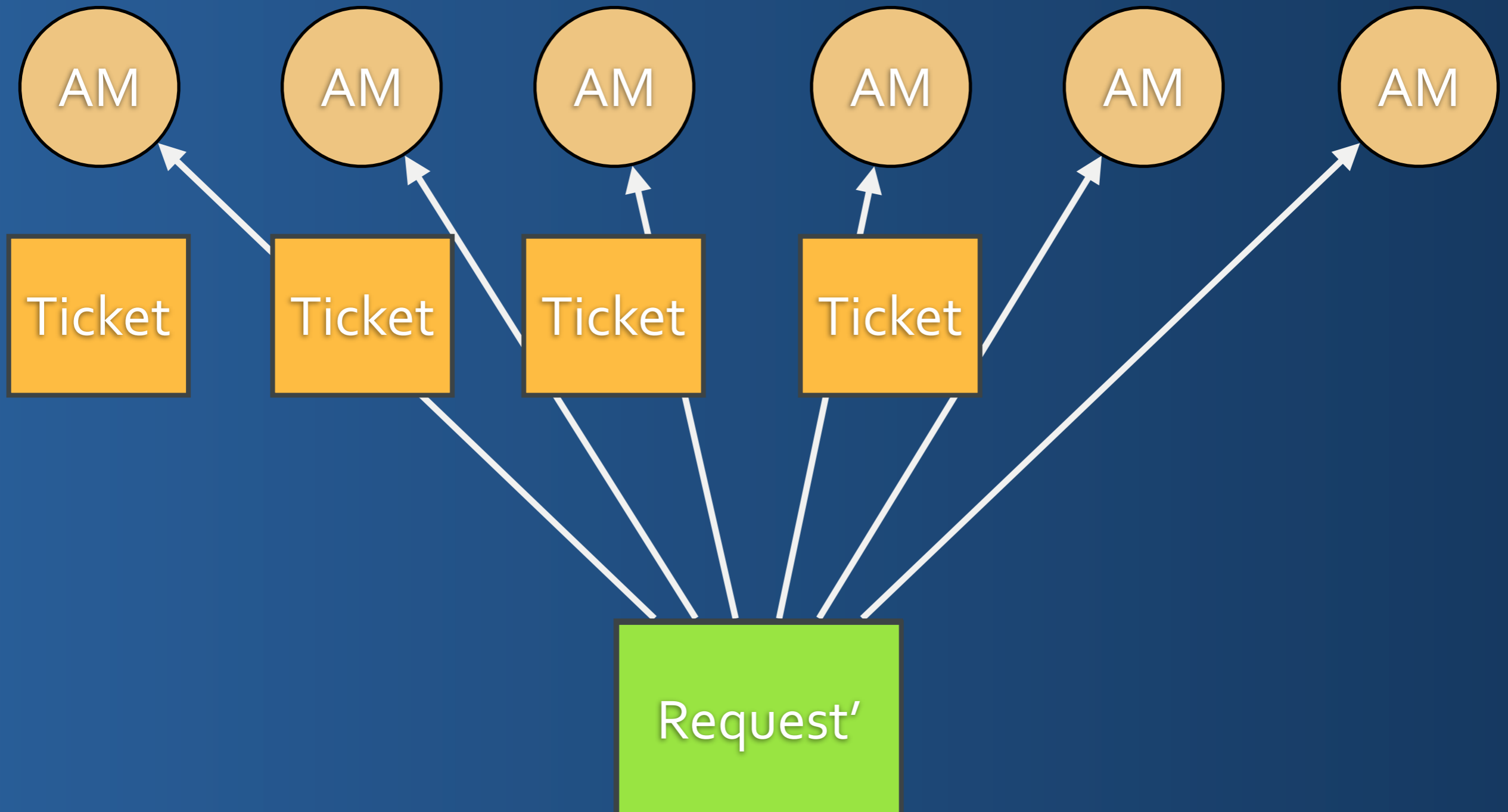
RSpec Lifecycle



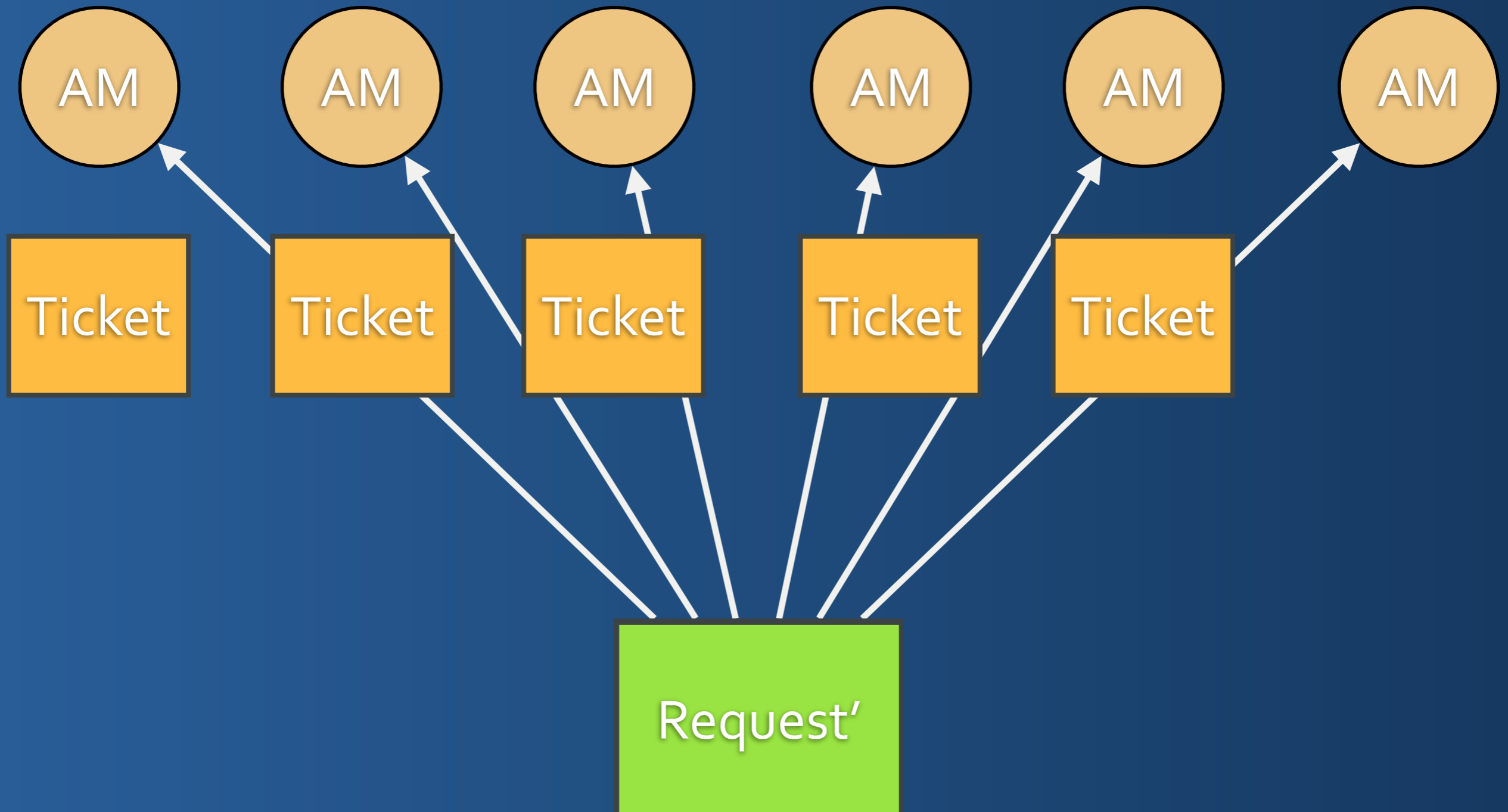
RSpec Lifecycle



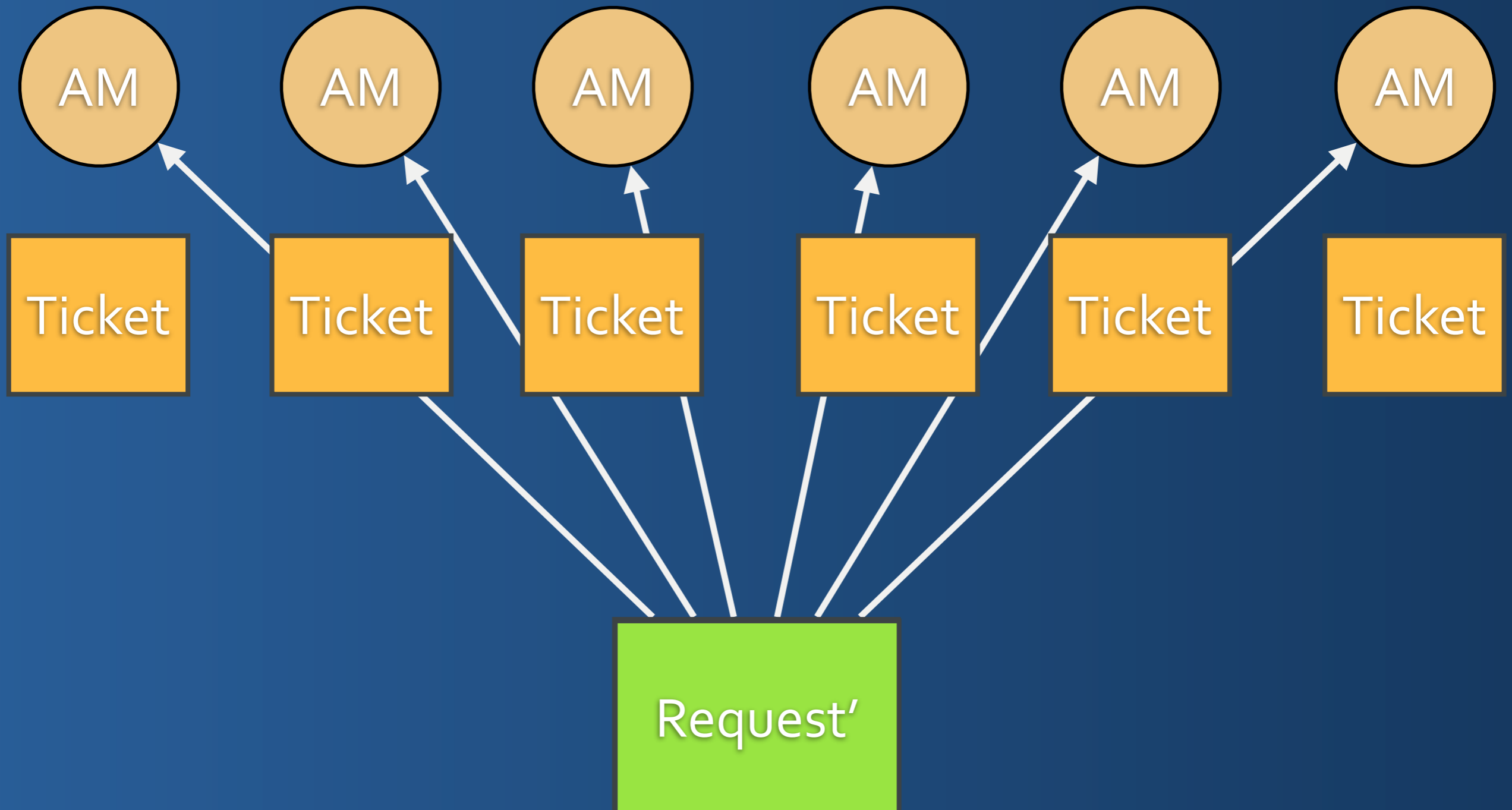
RSpec Lifecycle



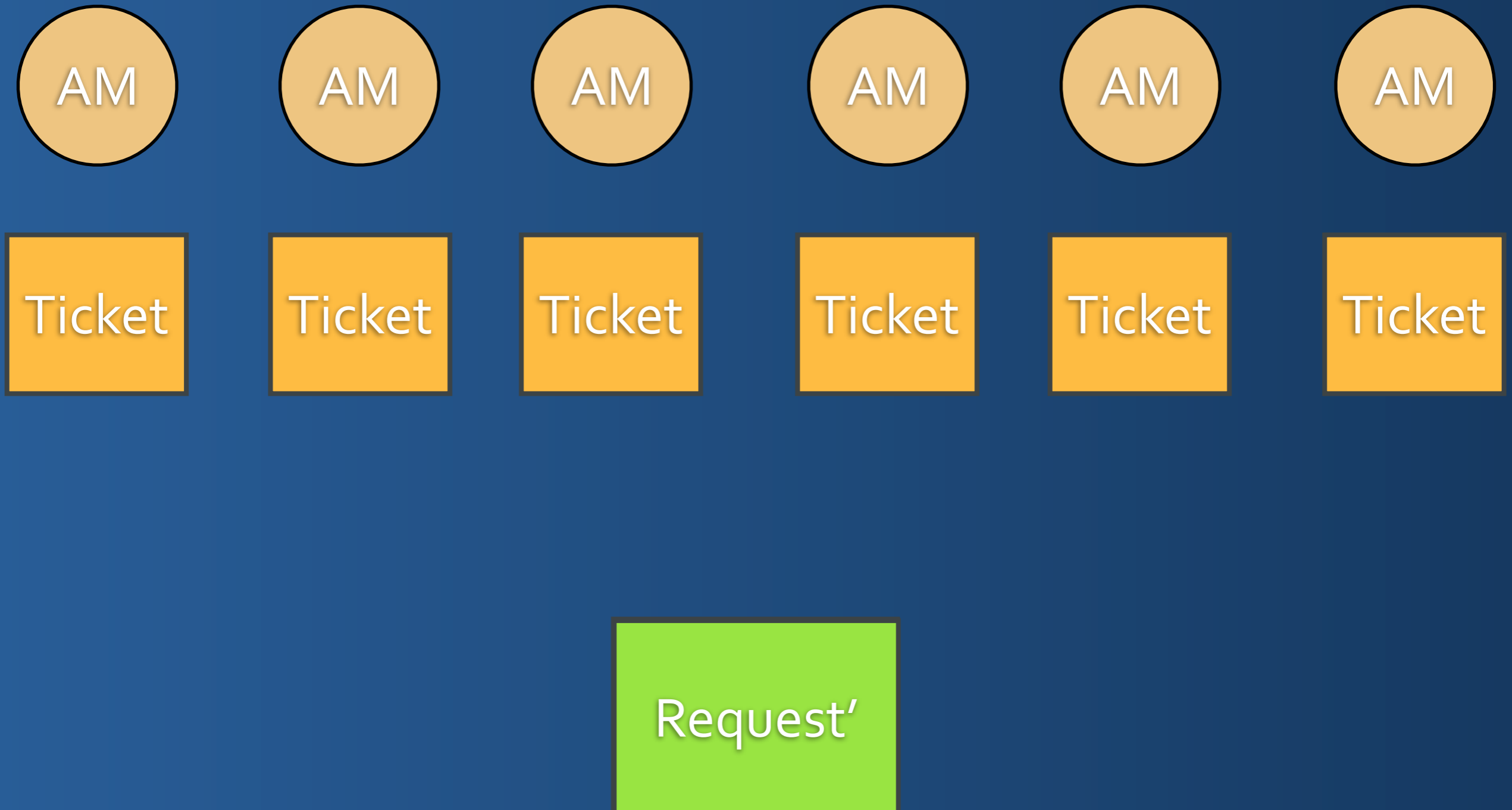
RSpec Lifecycle



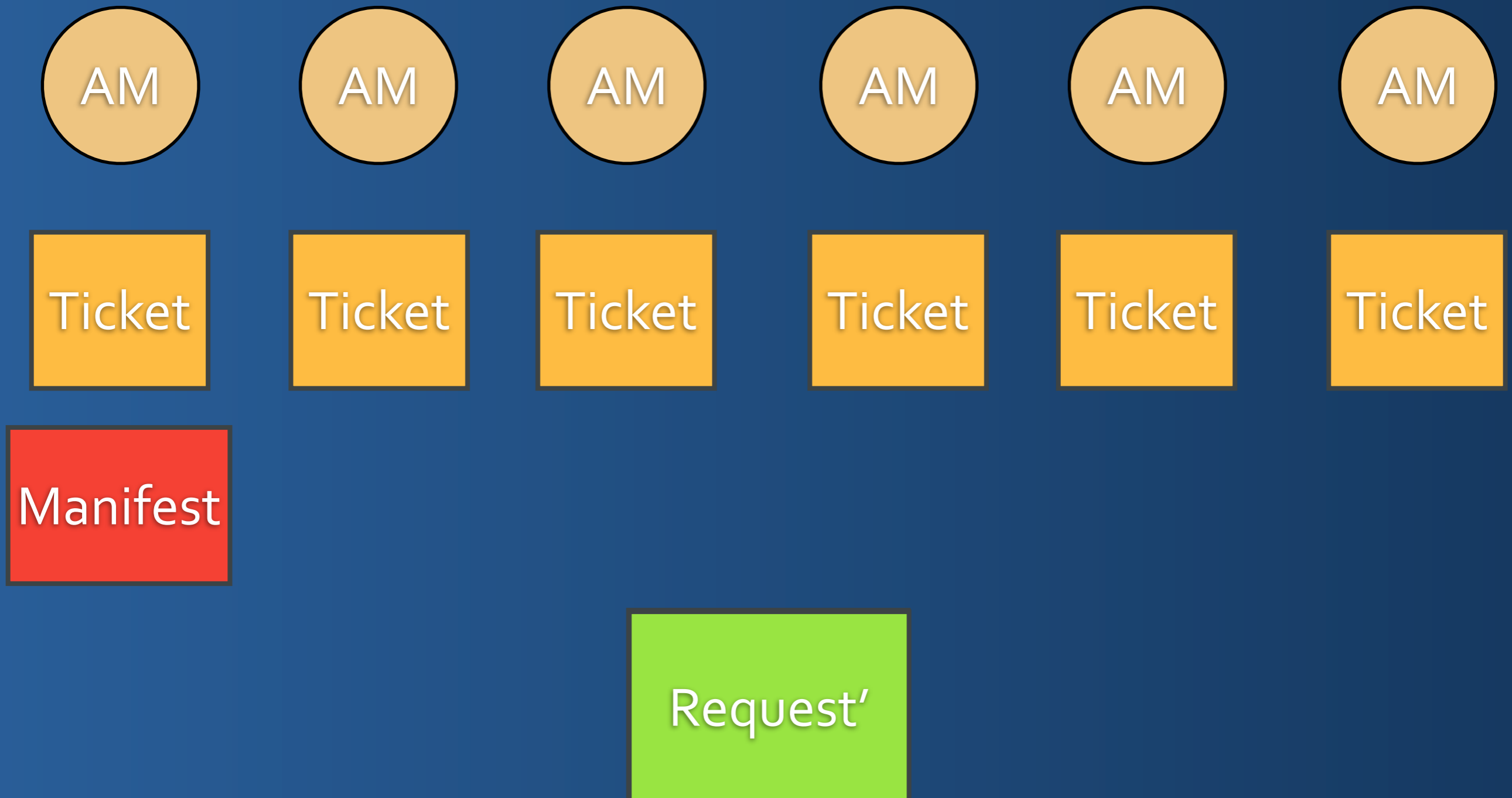
RSpec Lifecycle



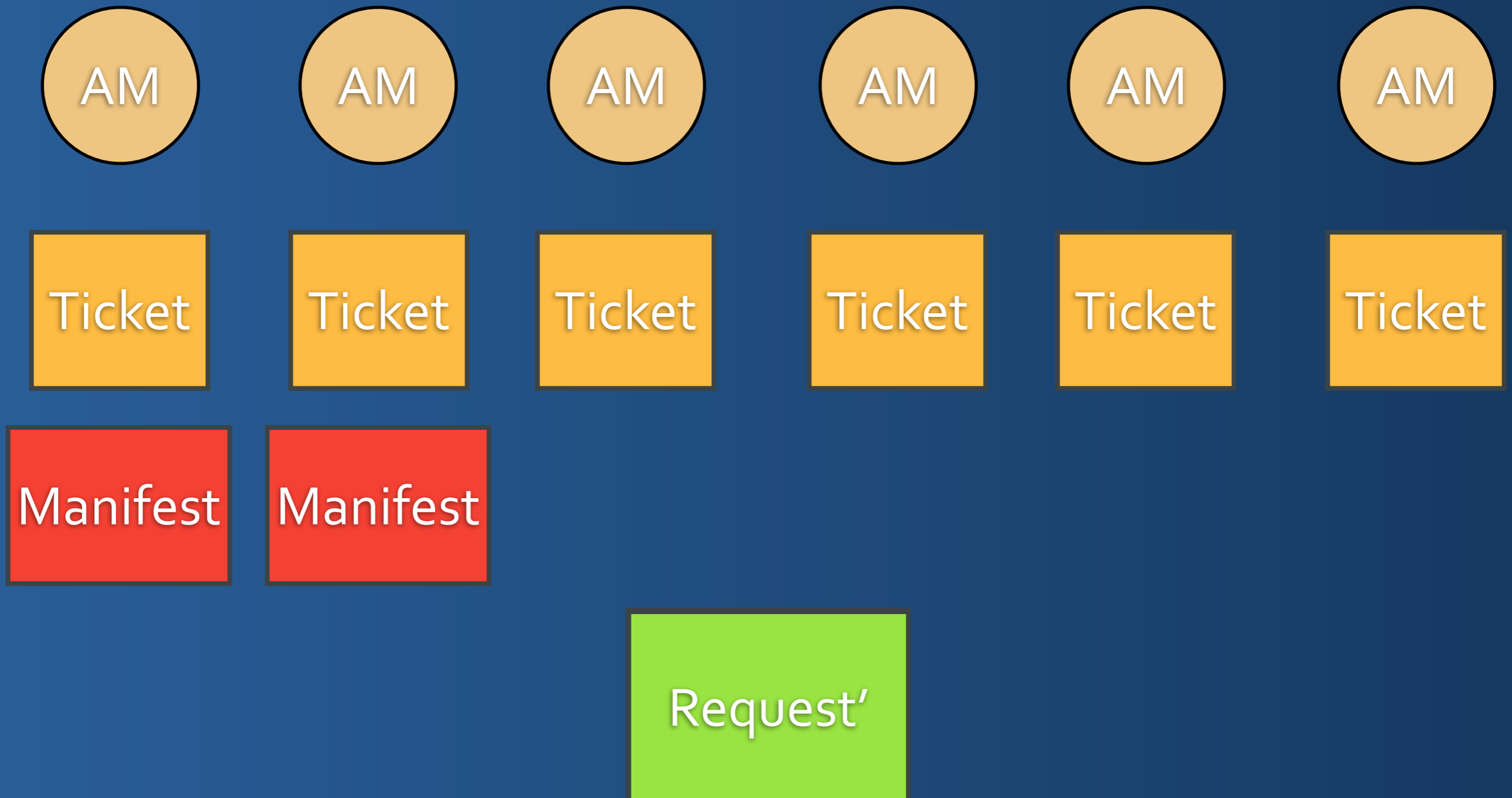
RSpec Lifecycle



RSpec Lifecycle



RSpec Lifecycle



RSpec Lifecycle



RSpec Lifecycle



RSpec Lifecycle



RSpec Lifecycle



Identifiers

- Using URNs proposed by GMOC
- Three types of identifiers
 - Component IDs
 - Virtual IDs (not URNs)
 - Sliver IDs

	Advertisement	Request	Ticket	Manifest
Component	Required	Optional	Required	Required
Virtual		Required	Required	Required
Sliver				Required

Extension Principles

- Lots of people are adding interesting stuff to ProtoGENI
 - I don't want to get in their way!
- If you don't know what it means, you probably don't want it!
- Specifying semantics as part of schema only gets you so far (cf. OpenFlow, BGP Mux)
- Hang extensions from anywhere
 - RSpec-wide, nodes, links, interfaces

Extension Mechanisms

- Using XML namespaces: each extension in a namespace
- Extension *may* be defined and validated by its own schema
- Location within parent schema done by convention

Interoperability Concerns

- Main need for interoperability:
 - Describing “touch points” between substrates
- “External references”
 - Think “extern” in C
 - In ProtoGENI model, touch points are always at interfaces
 - References to interface declarations in other RSpecs

Future of the PGENI RSpec

- Updates to core RSpec: Rare
- Lots and lots of extensions
 - Occasional standardization for the most important ones

www.protogeni.net