

# Atlas and Hestia

## Dependency Visualization and Management with Kotlin/JS

KotlinConf 2018-10-04

**Neil Power**

Senior Software Developer

[neil.power@hootsuite.com](mailto:neil.power@hootsuite.com)



# Is It Worth the Time?

HOW LONG CAN YOU WORK ON MAKING A ROUTINE TASK MORE EFFICIENT BEFORE YOU'RE SPENDING MORE TIME THAN YOU SAVE?  
(ACROSS FIVE YEARS)

		HOW OFTEN YOU DO THE TASK					
		50/DAY	5/DAY	DAILY	WEEKLY	MONTHLY	YEARLY
HOW MUCH TIME YOU SHAVE OFF	1 SECOND	1 DAY	2 HOURS	30 MINUTES	4 MINUTES	1 MINUTE	5 SECONDS
	5 SECONDS	5 DAYS	12 HOURS	2 HOURS	21 MINUTES	5 MINUTES	25 SECONDS
	30 SECONDS	4 WEEKS	3 DAYS	12 HOURS	2 HOURS	30 MINUTES	2 MINUTES
	1 MINUTE	8 WEEKS	6 DAYS	1 DAY	4 HOURS	1 HOUR	5 MINUTES
	5 MINUTES	9 MONTHS	4 WEEKS	6 DAYS	21 HOURS	5 HOURS	25 MINUTES
	30 MINUTES		6 MONTHS	5 WEEKS	5 DAYS	1 DAY	2 HOURS
	1 HOUR		10 MONTHS	2 MONTHS	10 DAYS	2 DAYS	5 HOURS
	6 HOURS				2 MONTHS	2 WEEKS	1 DAY
1 DAY					8 WEEKS	5 DAYS	

Source: <https://xkcd.com/1205/>



# Outline

- Why am I here?
- Background
- How We Work
- A Brief History of Android Development at Hootsuite
- Atlas
- Hestia
- Working with Kotlin/JS
- Tool Extensibility
- Conclusion



**Why am I here?**



# Speakers



**Neil Power**

I was an Android Developer at Hootsuite from 2015-2017.

Since 2017, I have been working Developer Products team, building out the Hootsuite Developer Platform.

My daily experience with Kotlin is now in tools such as Atlas, Hestia which I will be talking about today, as well as Hermes.



# Why am I here?

- Experimentation
  - Hootsuite was an early adopter of Kotlin in production.
  - While growing our Kotlin skills, we developed some in house tools written in Kotlin.



# Why am I here?

- Work out loud
  - Work passionately, know that what you're doing will be valuable to someone, and strive to share what you're working on with those around you.



# Background



# Background

Last year, I spoke at KotlinConf about our experience adopting Kotlin at Hootsuite.

As I don't think that Blizzard would want us using Peon images from Warcraft, we have renamed "Peon" to "Hestia"



# Background

Hestia is a goddess of Architecture, which makes it a fitting name for the tool.

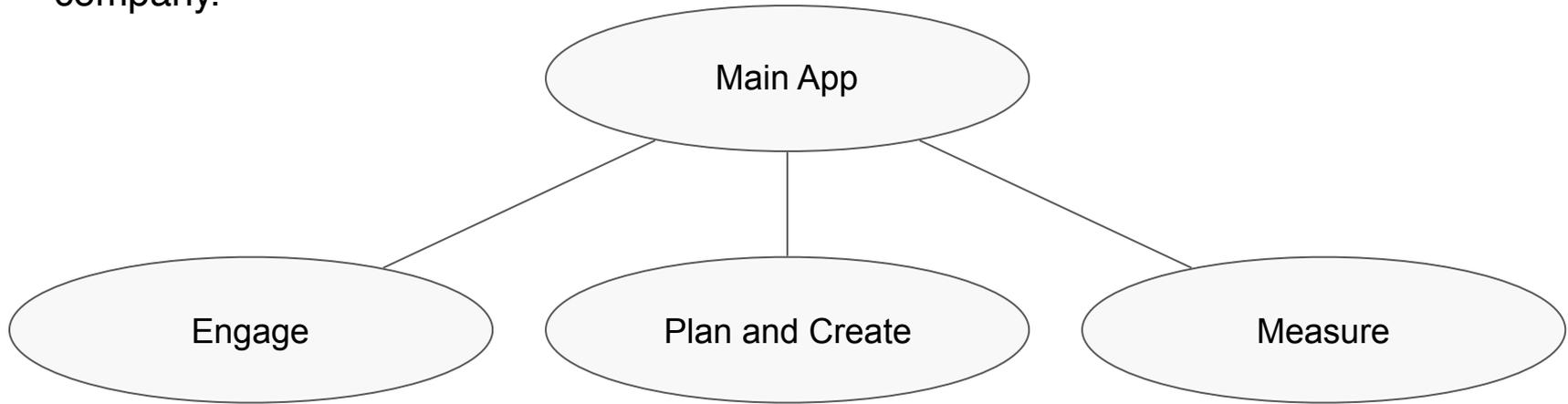


# How We Work



# How We Work

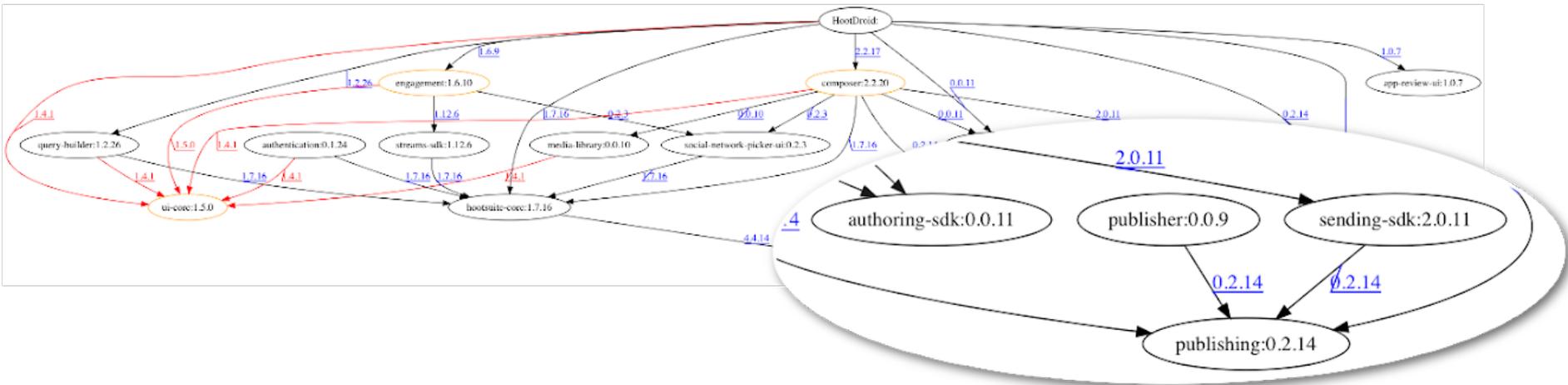
At Hootsuite, we have a core Mobile team focussed on core app functionality and on splitting the app into “verticals”. These verticals are worked on by other teams around the company.





# How we Work

In this dependency graph, each node is a repository that one or more teams contributes to and maintains.



# How we Work

In addition to Android development, we also practice library driven development on our iOS teams writing Swift, as well as our backend services teams writing Scala and Go.



# **A Brief History of Android Development at Hootsuite**



# Decomposition into Libraries

- Late 2015
  - We started to think about extracting functionality into Libraries
  - Each library is its own repository



# Decomposition into Libraries

- 2016
  - New features starting to be in their own libraries
  - Tools extracted into libraries
  - The start of Kotlin



# Decomposition into Libraries

- 2017
  - Many libraries (mostly Kotlin)
  - Ownership distributed across teams



# Decomposition into Libraries

- What?
  - New features in their own libraries.
  - Decompose HootDroid into reusable library components.

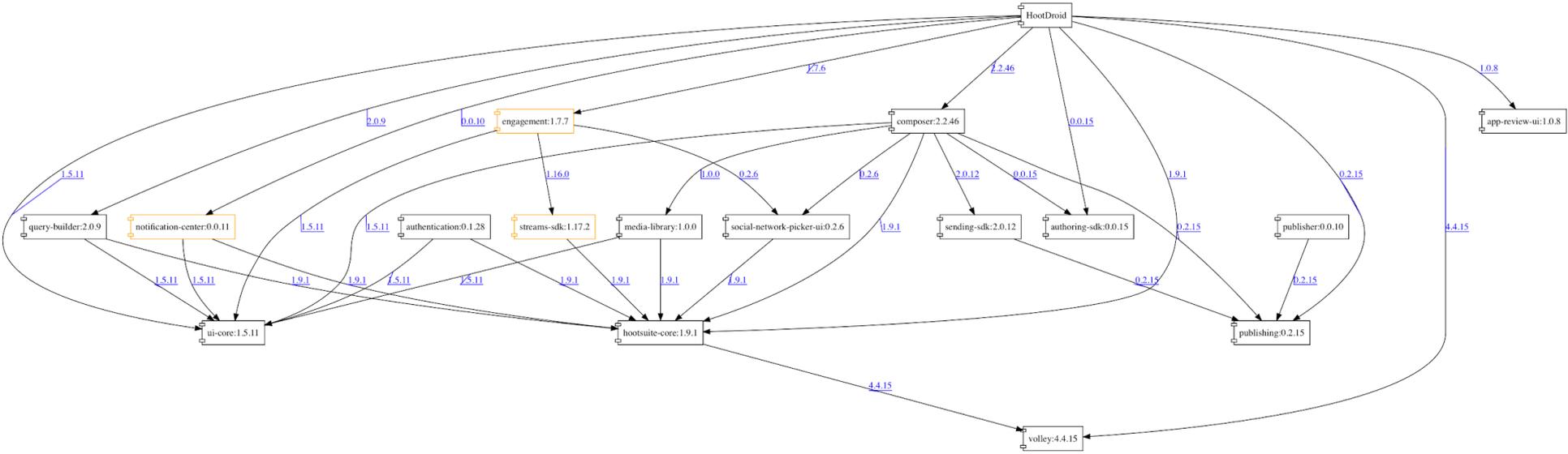


# Decomposition into Libraries

- Why?
  - Better Architecture.
    - Separating libraries forces you to consider interfaces between features and main app.
  - Historically, it allowed us to Introduce Kotlin in a steady, but controlled way.



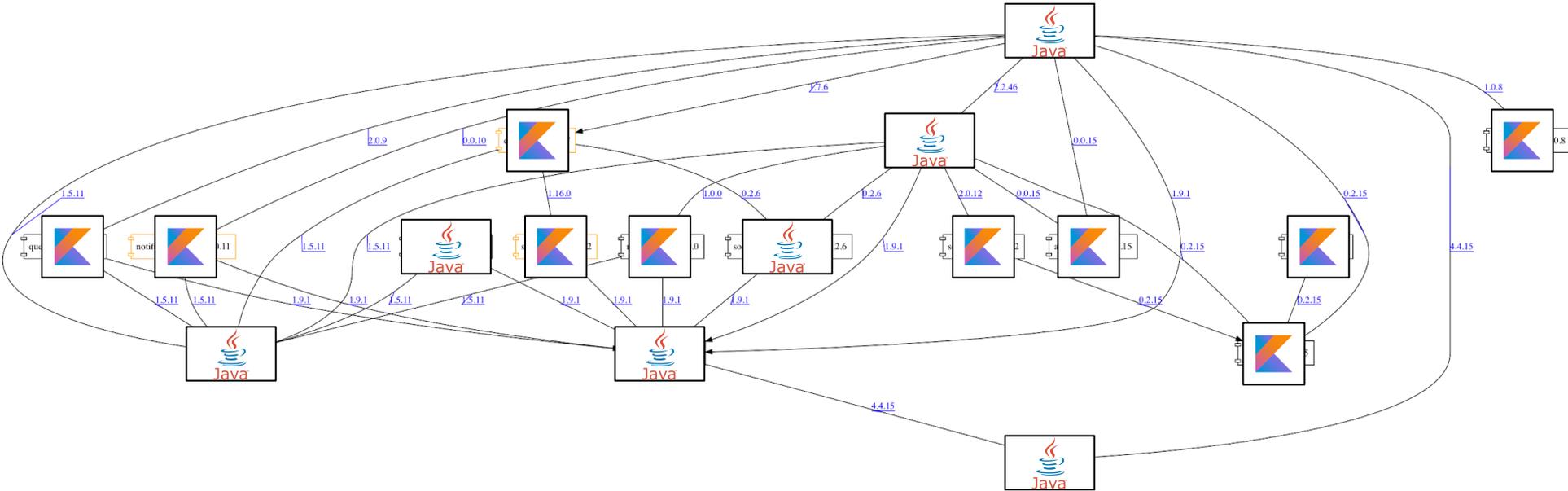
# Library Dependency Graph





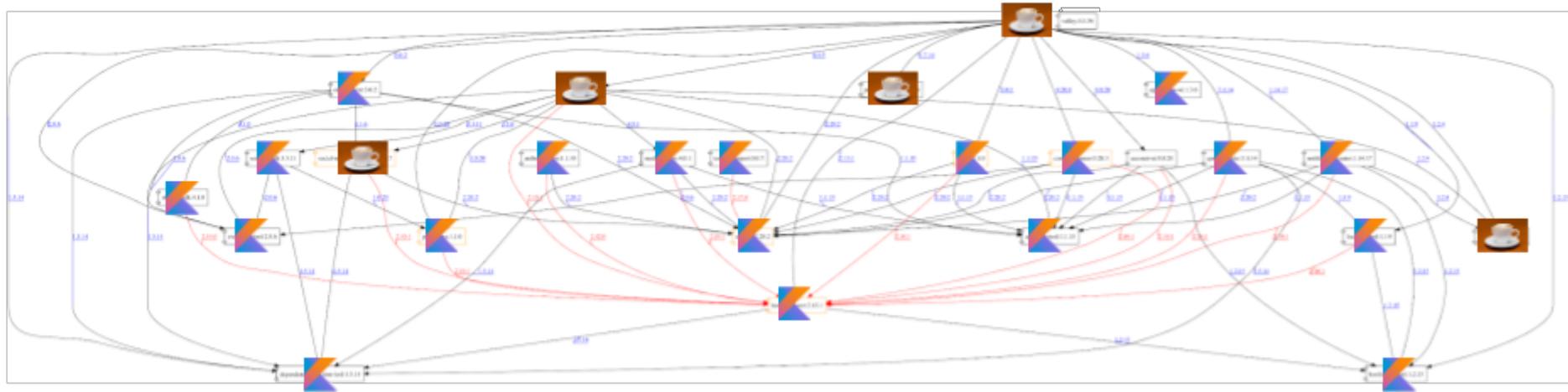


# Library Dependency Graph





# Library Dependency Graph



# Decomposition into Libraries

- Benefits
  - Easy to make new libraries Kotlin only.
  - Easy to delegate ownership to different teams.
  - Faster builds when using precompiled modules.



# Decomposition into Libraries

- Drawbacks
  - Increased overhead for project configuration.
  - Need to keep libraries versioned and in sync.



# Tools

- Library Driven Development creates some development overhead.
  - Who wants to manually bump versions of internal libraries?
  - At Hootsuite, we use Atlas and Hestia to mitigate this overhead.



# Atlas



# What is Atlas?

- Atlas crawls one or more github hosts (github.com / private Github Enterprise) looking through organizations or repos.



# What is Atlas?

- Atlas crawls one or more github hosts (github.com / private Github Enterprise) looking through organizations or repos.
- Atlas parses and reads dependency files
  - Gradle files for Android
  - Carthage files for iOS
  - SBT files for Scala



# What is Atlas?

- Atlas crawls one or more github hosts (github.com / private Github Enterprise) looking through organizations or repos.
- Atlas parses and reads dependency files
  - Gradle files for Android
  - Carthage files for iOS
  - SBT files for Scala
- From those dependencies, Atlas generates a graph which can either be visualized or used as a data structure.



# Atlas

- Atlas started as a command line Ruby application which determines project dependencies. It is now written in Kotlin/JS.



# Atlas

- Atlas started as a command line Ruby application which determines project dependencies. It is now written in Kotlin/JS.
- We use it to create dependency graphs for iOS, Android, and Scala projects.



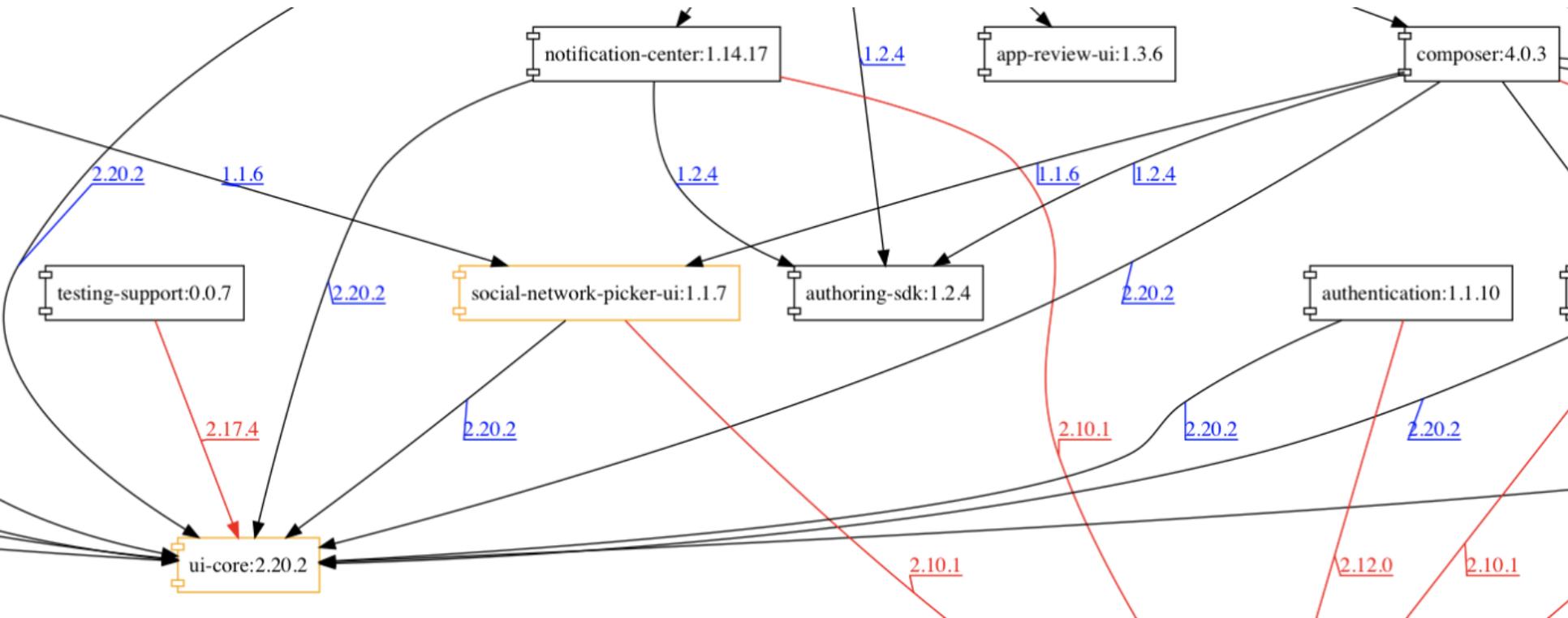
# Atlas

- Atlas started as a command line Ruby application which determines project dependencies. It is now written in Kotlin/JS.
- We use it to create dependency graphs for iOS, Android, and Scala projects.
- Atlas can visualize the graphs to help developers understand their projects better.

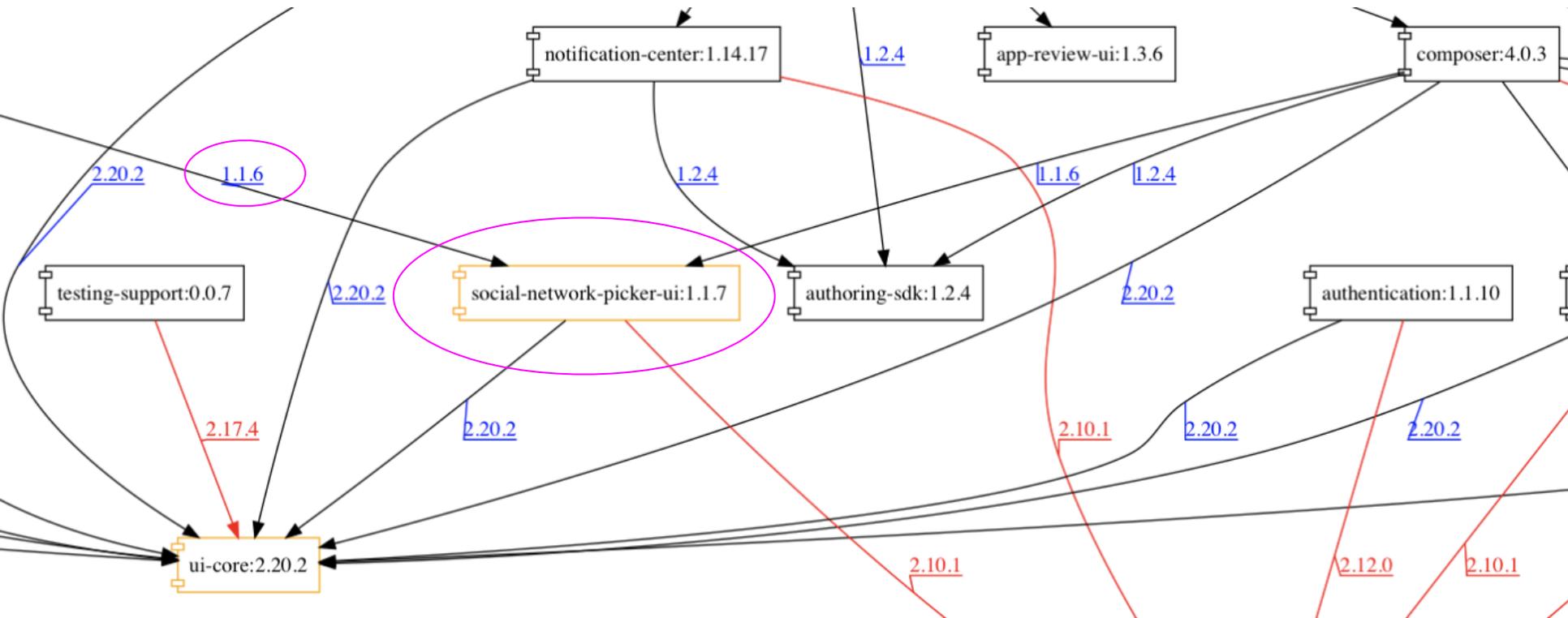




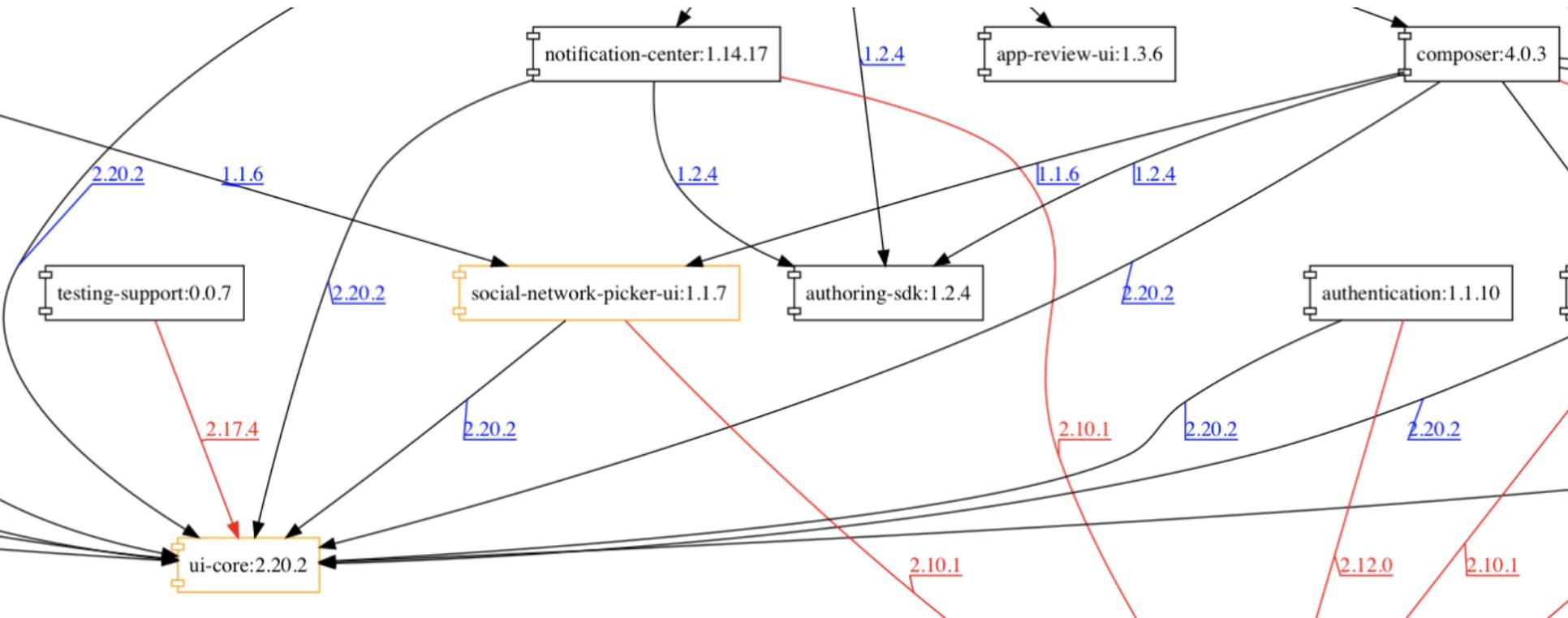
# Atlas



# Atlas



# Atlas



# Atlas

- What do we use Atlas for?
  - Onboarding new Android Developers.



# Atlas

- What do we use Atlas for?
  - Onboarding new Android Developers.
  - Atlas graphs are shown on status boards in the office.



# Atlas

- What do we use Atlas for?
  - Onboarding new Android Developers.
  - Atlas graphs are shown on status boards in the office.
  - Diagnose bugs in development.



# Atlas

- What do we use Atlas for?
  - Onboarding new Android Developers.
  - Atlas graphs shown on status boards in the office.
  - Diagnose bugs in development.
  - Graph building and sorting for Hestia.



# Atlas

- What do we use Atlas for?
  - Onboarding new Android Developers.
  - Atlas graphs shown on status boards in the office.
  - Diagnose bugs in development.
  - Graph building and sorting for Hestia.

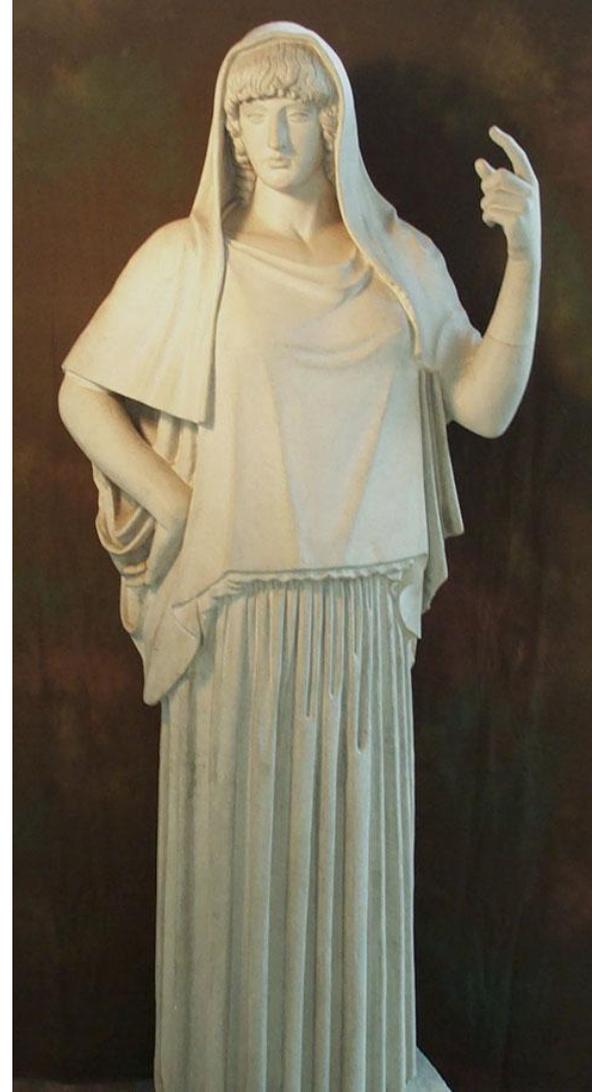


# Hestia



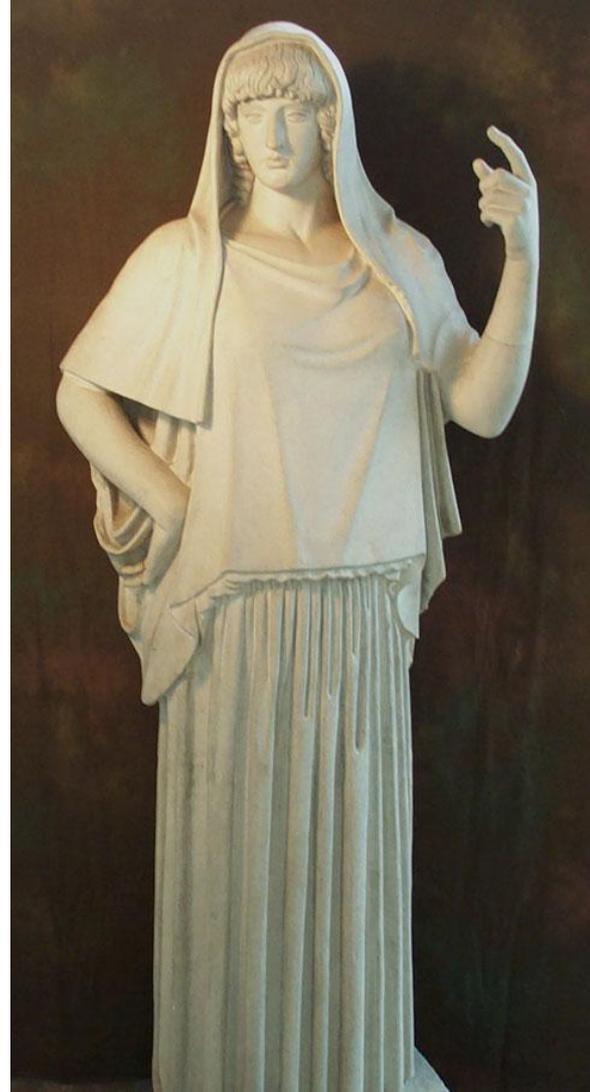
# Hestia

- Hestia is another command line Ruby application, rewritten in Kotlin/JS.



# Hestia

- Hestia is another command line Ruby application, rewritten in Kotlin/JS.
- It takes as input, the graph data structure that Atlas outputs as well as dependency updates you wish to apply.

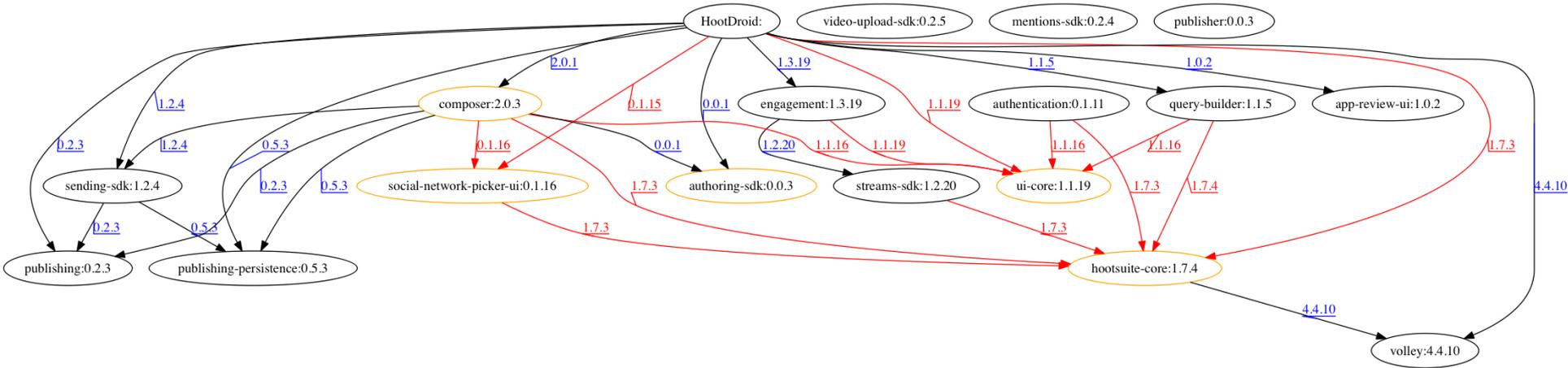


# Hestia

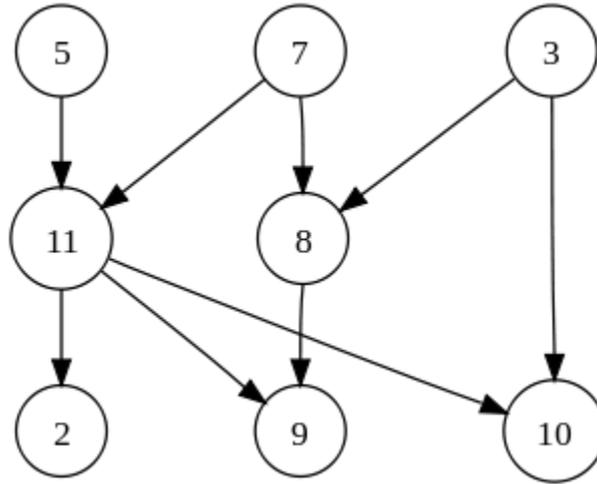
- Hestia is another command line Ruby application, rewritten in Kotlin/JS.
- It takes as input, the graph data structure that Atlas outputs as well as dependency updates you wish to apply.
- Hestia then topologically sorts the graph and creates ordered pull requests with the given changes.



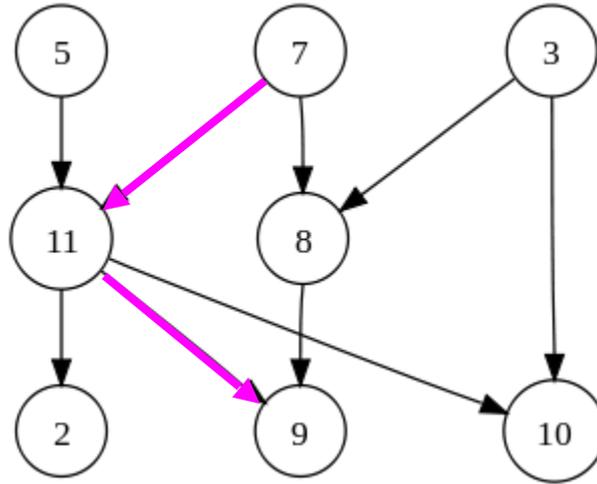
# Minor Version Changes



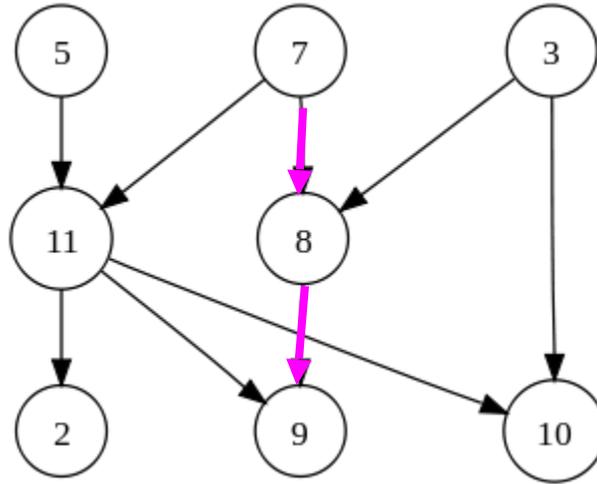
# Topological Sort



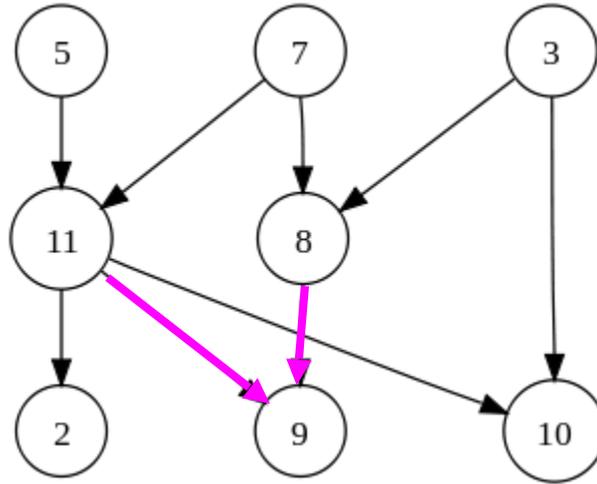
# Topological Sort



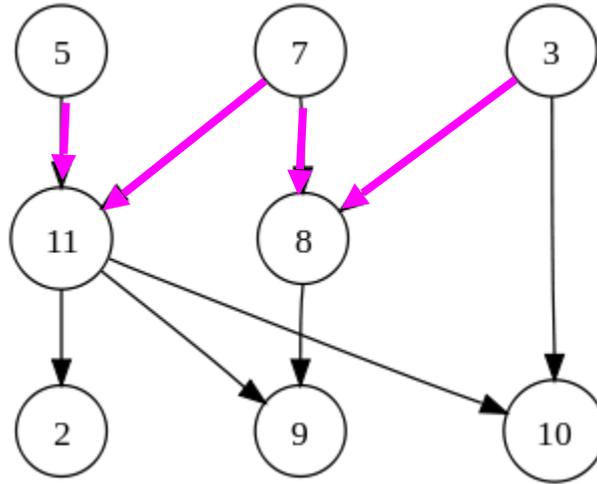
# Topological Sort



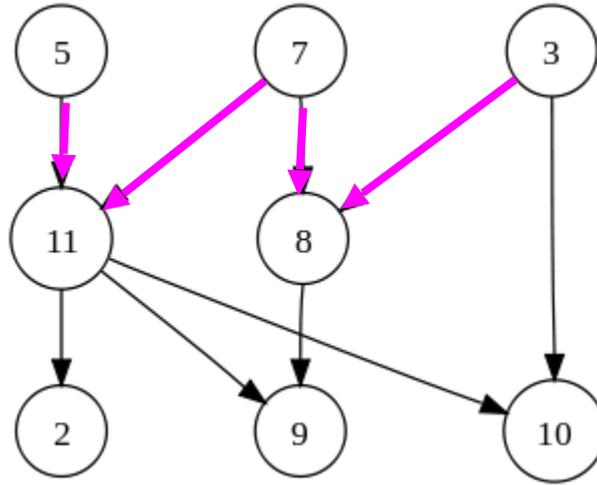
# Topological Sort



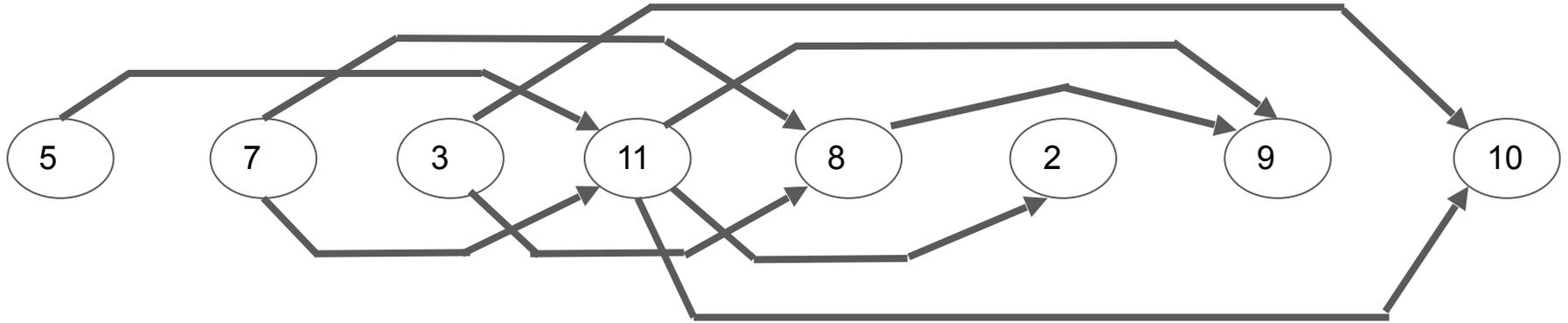
# Topological Sort



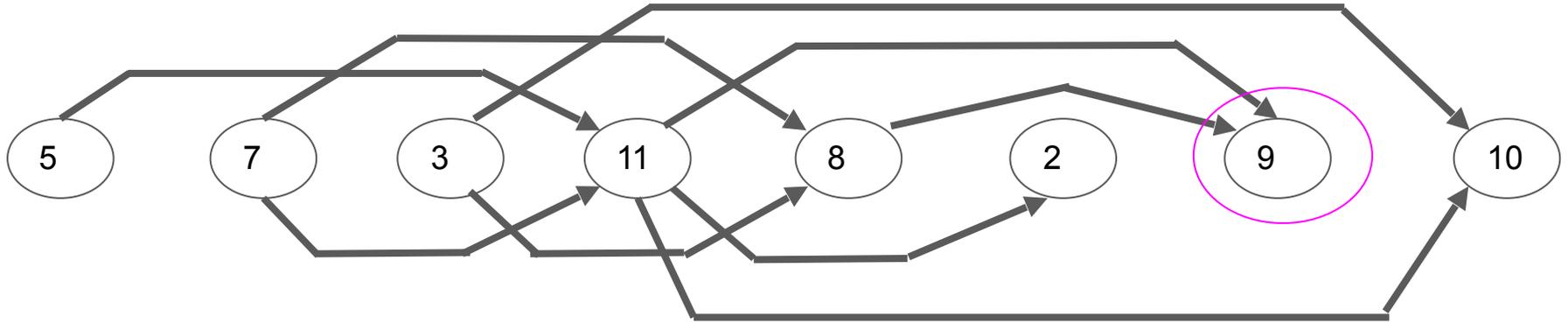
# Topological Sort



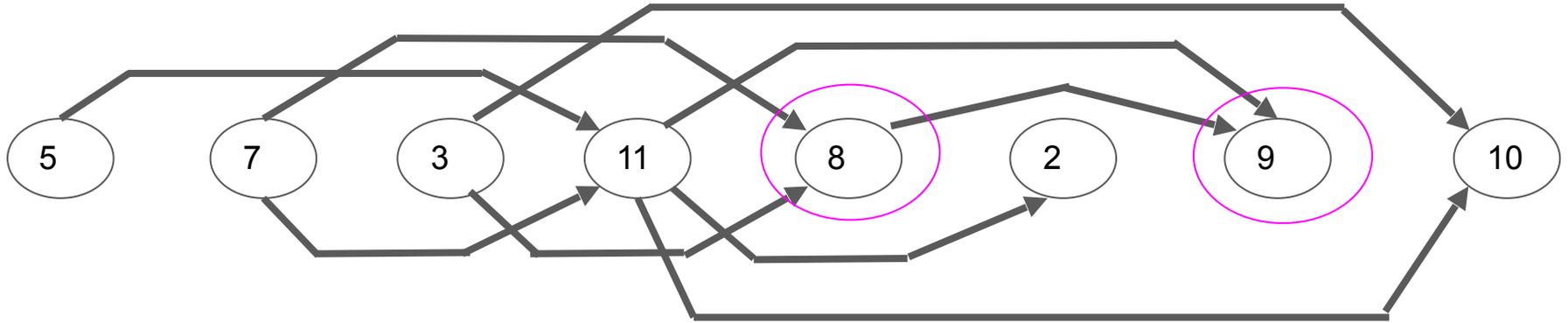
# Topological Sort



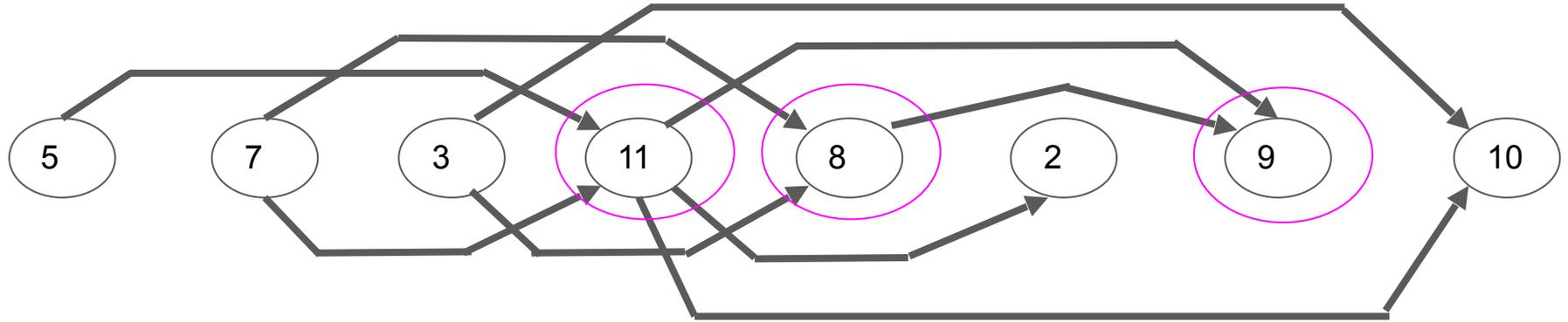
# Topological Sort



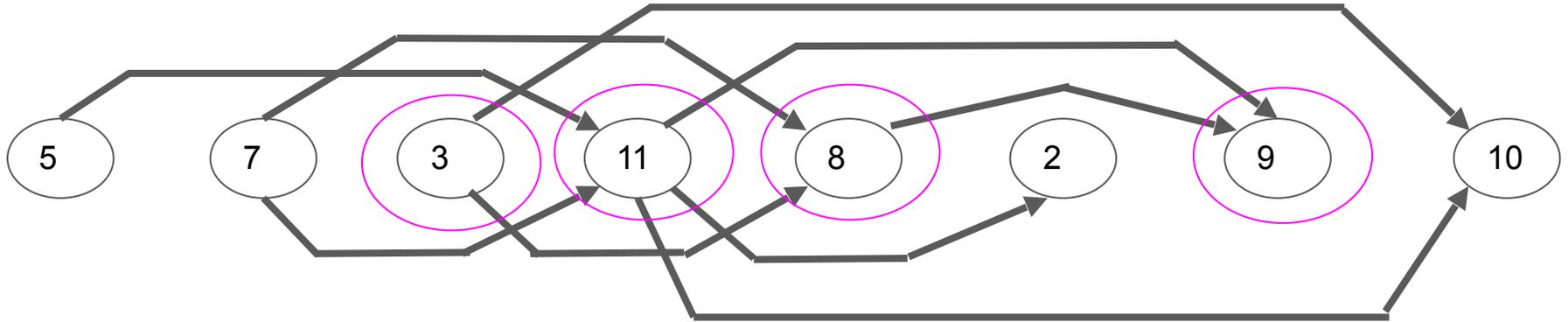
# Topological Sort



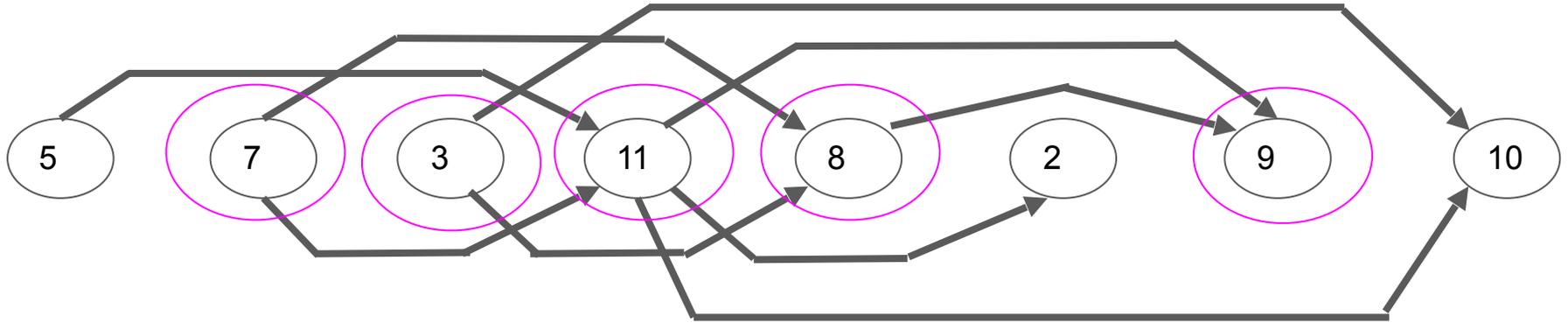
# Topological Sort



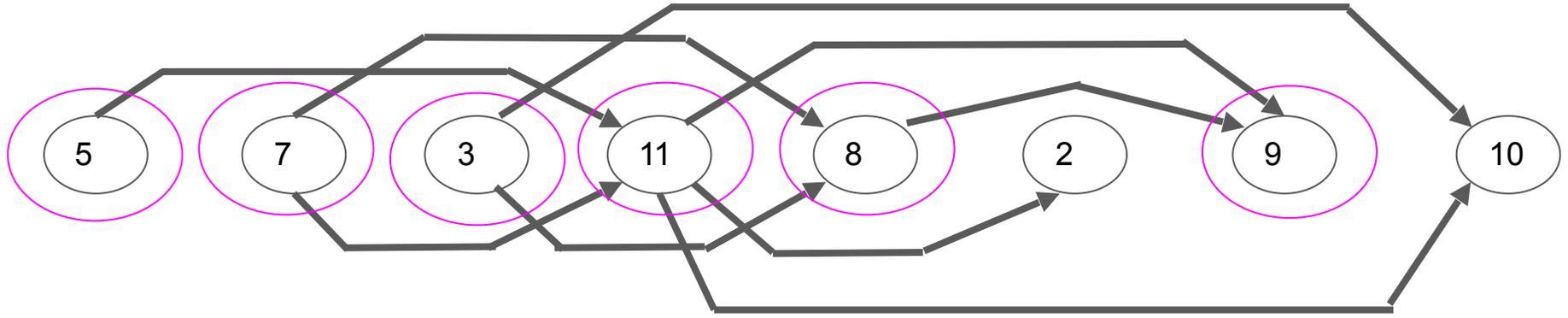
# Topological Sort



# Topological Sort



# Topological Sort





peon APP 1:06 PM ☆

Job's done: Kotlin 1.1.0

- 1 - com.hootsuite.android:event-bus-tool:1.0.3 - <https://github.hootops.com/Android/event-bus-tool/pull/10>
- 2 - com.hootsuite.android:dependency-injection-tool:1.2.3 - <https://github.hootops.com/Android/dependency-injection-tool/pull/19>
- 3 - com.hootsuite.android:legacy:volley:4.4.11 - <https://github.hootops.com/Android/Volley/pull/13>
- 4 - com.hootsuite.android:hootlogger-tool:1.0.5 - <https://github.hootops.com/Android/hootlogger-tool/pull/13>
- 5 - com.hootsuite.android:hootsuite-core:1.7.9 - <https://github.hootops.com/Android/hootsuite-core/pull/59>
- 6 - com.hootsuite.android:streams-sdk:1.5.3 - <https://github.hootops.com/Android/streams-sdk/pull/141>
- 7 - com.hootsuite.android:ui-core:1.3.5 - <https://github.hootops.com/Android/ui-core/pull/96>
- 8 - com.hootsuite.android:analytics-tool:1.0.3 - <https://github.hootops.com/Android/analytics-tool/pull/10>
- 9 - com.hootsuite.android:engagement:1.3.50 - <https://github.hootops.com/Android/engagement/pull/198>
- 10 - com.hootsuite.android:social-network-picker-ui:0.1.19 - <https://github.hootops.com/Android/social-network-picker-ui/pull/49>
- 11 - com.hootsuite.android:publishing:0.2.7 - <https://github.hootops.com/Android/publishing/pull/18>
- 12 - com.hootsuite.android:sending-sdk:2.0.4 - <https://github.hootops.com/Android/sending-sdk/pull/56>
- 13 - com.hootsuite.android:authoring-sdk:0.0.6 - <https://github.hootops.com/Android/authoring-sdk/pull/8>
- 14 - com.hootsuite.android:composer:2.1.11 - <https://github.hootops.com/Android/compose/pull/222>
- 15 - com.hootsuite.android:purchasing-tool:1.6.3 - <https://github.hootops.com/Android/purchasing-tool/pull/16>
- 16 - com.hootsuite.android:app-review-ui:1.0.3 - <https://github.hootops.com/Android/app-review-ui/pull/9>
- 17 - com.hootsuite.android:location-tool:0.1.1 - <https://github.hootops.com/Android/location-tool/pull/5>
- 18 - com.hootsuite.android:query-builder:1.2.5 - <https://github.hootops.com/Android/query-builder/pull/204>
- 19 - Root Level Change - <https://github.hootops.com/Android/HootDroid/pull/1239>
- 20 - com.hootsuite.android:authentication:0.1.18 - <https://github.hootops.com/Android/authentication/pull/30>
- 21 - com.hootsuite.android:publisher:0.0.4 - <https://github.hootops.com/Android/publisher/pull/4>



peon APP 1:06 PM ☆

Job's done: Kotlin 1.1.0

- 1 - com.hootsuite.android:event-bus-tool:1.0.3 - <https://github.hootops.com/Android/event-bus-tool/pull/10>
- 2 - com.hootsuite.android:dependency-injection-tool:1.2.3 - <https://github.hootops.com/Android/dependency-injection-tool/pull/19>
- 3 - com.hootsuite.android:legacy:volley:4.4.11 - <https://github.hootops.com/Android/Volley/pull/13>
- 4 - com.hootsuite.android:hootlogger-tool:1.0.5 - <https://github.hootops.com/Android/hootlogger-tool/pull/13>
- 5 - com.hootsuite.android:hootsuite-core:1.7.9 - <https://github.hootops.com/Android/hootsuite-core/pull/59>
- 6 - com.hootsuite.android:streams-sdk:1.5.3 - <https://github.hootops.com/Android/streams-sdk/pull/141>
- 7 - com.hootsuite.android:ui-core:1.3.5 - <https://github.hootops.com/Android/ui-core/pull/96>
- 8 - com.hootsuite.android:analytics-tool:1.0.3 - <https://github.hootops.com/Android/analytics-tool/pull/10>
- 9 - com.hootsuite.android:engagement:1.3.50 - <https://github.hootops.com/Android/engagement/pull/198>
- 10 - com.hootsuite.android:social-network-picker-ui:0.1.19 - <https://github.hootops.com/Android/social-network-picker-ui/pull/49>
- 11 - com.hootsuite.android:publishing:0.2.7 - <https://github.hootops.com/Android/publishing/pull/18>
- 12 - com.hootsuite.android:sending-sdk:2.0.4 - <https://github.hootops.com/Android/sending-sdk/pull/56>
- 13 - com.hootsuite.android:authoring-sdk:0.0.6 - <https://github.hootops.com/Android/authoring-sdk/pull/8>
- 14 - com.hootsuite.android:composer:2.1.11 - <https://github.hootops.com/Android/compose/pull/222>
- 15 - com.hootsuite.android:purchasing-tool:1.6.3 - <https://github.hootops.com/Android/purchasing-tool/pull/16>
- 16 - com.hootsuite.android:app-review-ui:1.0.3 - <https://github.hootops.com/Android/app-review-ui/pull/9>
- 17 - com.hootsuite.android:location-tool:0.1.1 - <https://github.hootops.com/Android/location-tool/pull/5>
- 18 - com.hootsuite.android:query-builder:1.2.5 - <https://github.hootops.com/Android/query-builder/pull/204>
- 19 - Root Level Change - <https://github.hootops.com/Android/HootDroid/pull/1239>
- 20 - com.hootsuite.android:authentication:0.1.18 - <https://github.hootops.com/Android/authentication/pull/30>
- 21 - com.hootsuite.android:publisher:0.0.4 - <https://github.hootops.com/Android/publisher/pull/4>

```
4     targetSdkVersion = 24
5     buildToolsVersion = "24.0.1"
6     supportLibVersion = "24.1.0"
7 +   kotlinVersion = "1.1.0"
8   }
9
10  buildscript {
```

View



```
1 +VERSION_NAME=1.0.3
2
3 POM_ARTIFACT_ID=event-bus-tool
4 GROUP=com.hootsuite.android
```



peon APP 1:06 PM ☆

Job's done: Kotlin 1.1.0

- 1 - com.hootsuite.android:event-bus-tool:1.0.3 - <https://github.hootops.com/Android/event-bus-tool/pull/10>
- 2 - com.hootsuite.android:dependency-injection-tool:1.2.3 - <https://github.hootops.com/Android/dependency-injection-tool/pull/19>
- 3 - com.hootsuite.android.legacy:volley:4.4.11 - <https://github.hootops.com/Android/Volley/pull/13>
- 4 - com.hootsuite.android:hootlogger-tool:1.0.5 - <https://github.hootops.com/Android/hootlogger-tool/pull/13>
- 5 - com.hootsuite.android:hootsuite-core:1.7.9 - <https://github.hootops.com/Android/hootsuite-core/pull/59>
- 6 - com.hootsuite.android:streams-sdk:1.5.3 - <https://github.hootops.com/Android/streams-sdk/pull/141>
- 7 - com.hootsuite.android:ui-core:1.3.5 - <https://github.hootops.com/Android/ui-core/pull/96>
- 8 - com.hootsuite.android:analytics-tool:1.0.3 - <https://github.hootops.com/Android/analytics-tool/pull/10>
- 9 - com.hootsuite.android:engagement:1.3.50 - <https://github.hootops.com/Android/engagement/pull/198>
- 10 - com.hootsuite.android:social-network-picker-ui:0.1.19 - <https://github.hootops.com/Android/social-network-picker-ui/pull/49>
- 11 - com.hootsuite.android:publishing:0.2.7 - <https://github.hootops.com/Android/publishing/pull/18>
- 12 - com.hootsuite.android:sending-sdk:2.0.4 - <https://github.hootops.com/Android/sending-sdk/pull/56>
- 13 - com.hootsuite.android:authoring-sdk:0.0.6 - <https://github.hootops.com/Android/authoring-sdk/pull/8>
- 14 - com.hootsuite.android:composer:2.1.11 - <https://github.hootops.com/Android/compose/pull/222>
- 15 - com.hootsuite.android:purchasing-tool:1.6.3 - <https://github.hootops.com/Android/purchasing-tool/pull/16>
- 16 - com.hootsuite.android:app-review-ui:1.0.3 - <https://github.hootops.com/Android/app-review-ui/pull/9>
- 17 - com.hootsuite.android:location-tool:0.1.1 - <https://github.hootops.com/Android/location-tool/pull/5>
- 18 - com.hootsuite.android:query-builder:1.2.5 - <https://github.hootops.com/Android/query-builder/pull/204>
- 19 - Root Level Change - <https://github.hootops.com/Android/HootDroid/pull/1239>
- 20 - com.hootsuite.android:authentication:0.1.18 - <https://github.hootops.com/Android/authentication/pull/30>
- 21 - com.hootsuite.android:publisher:0.0.4 - <https://github.hootops.com/Android/publisher/pull/4>

```
compile 'ca.barrenechea.header-decor:header-decor:0.2.6'

// Hootsuite Verticals (should wrap underlying business logic)
+ compile 'com.hootsuite.android:engagement:1.3.50'
+ compile('com.hootsuite.android:composer:2.1.11') {
    exclude group: 'com.android.support', module: 'appcompat-v7'
    exclude group: 'com.android.support', module: 'support-compat'
}

// Hootsuite dependencies
+ compile 'com.hootsuite.android:purchasing-tool:1.6.3'
+ compile 'com.hootsuite.android:publishing:0.2.7'
+ compile 'com.hootsuite.android:event-bus-tool:1.0.3'
+ compile 'com.hootsuite.android:dependency-injection-tool:1.2.3'
+ compile 'com.hootsuite.android:social-network-picker-ui:0.1.19'
+ compile 'com.hootsuite.android:supertooltips:4.0.0'
+ compile 'com.hootsuite.android:hootlogger-tool:1.0.5'
+ compile 'com.hootsuite.android:legacy:volley:4.4.11'
+ compile 'com.hootsuite.android:app-review-ui:1.0.3'
+ compile 'com.hootsuite.android:location-tool:0.1.1'

+ compile('com.hootsuite.android:hootsuite-core:1.7.9') {
    exclude group: 'com.android.support', module: 'appcompat-v7'
    exclude group: 'com.hootsuite.android:legacy', module: 'volley'
    exclude group: 'com.google.code.gson', module: 'gson'
    exclude group: 'org.apache.commons', module: 'commons-lang3'
}
+ compile('com.hootsuite.android:authoring-sdk:0.0.6') {
    exclude group: 'com.android.support', module: 'appcompat-v7'
}
+ compile('com.hootsuite.android:query-builder:1.2.5') {
    exclude group: 'com.android.support', module: 'appcompat-v7'
}
+ compile('com.hootsuite.android:ui-core:1.3.5') {
    exclude group: 'com.android.support', module: 'appcompat-v7' // Remember to update project.properties too!
}
}
```





# Project Hestia

This build requires parameters:

repo\_name

The name of the repo to update. Examples are engagement, streams-sdk, hootsuite-core, ui-core  
Cannot be used with the build\_all flag.

build\_all

When True, all the repos in the graph will be modified.  
Cannot be used with the repo argument or the version argument

commit\_message

The commit message to include for the change for each repo

jira\_ticket

The JIRA ticket reference. For example, IOS-1233, AND-3631, TA-121, etc

artifacts

List of artifacts separated by colon, in the format of: group\_id:artifact\_id:version. For example, com.hootsuite.android:engagement:1.5.0

properties

List of properties to be updated separated by colon, in the format of: property\_name:property\_value. For example, supportLibVersion:24.1.0

Options are:  
minSdkVersion  
compileSdkVersion  
targetSdkVersion  
buildToolsVersion  
supportLibVersion  
kotlinVersion  
gradleVersion  
daggerVersion

gradle

The gradle wrapper version. For example: 3.3

version

Version of the target library to update to. For Example: 1.2.3  
If not provided, the target library will have its patch version incremented only.

slack\_names

Space or Newline separated list of names to notify in slack, when a peon either fails or finishes.  
Use first names of team members



# Running Hestia

Input Argument	Example	Notes
Commit Message	-m "Updating Support Library"	
JIRA Ticket	-j AND-9999	
Repo Name	-r query-builder	Build up the graph from this repo
Org level change	-o	Update all the repos in the graph
Artifacts	-a com.hootsuite.android:query-builder:1.2.0	Update this artifact when encountered
Properties	-p minSdkVersion:17	Update this property when Encountered
Version	-v 9.9.9	Update the starting repo to this verison
Names	-n "neil simon"	Message in slack of PRs or Error

# Running Hestia

Input Argument	Example	Notes
Commit Message	-m "Updating Support Library"	
JIRA Ticket	-j AND-9999	
Repo Name	-r query-builder	Build up the graph from this repo
Org level change	-o	Update all the repos in the graph
Artifacts	-a com.hootsuite.android:query-builder:1.2.0	Update this artifact when encountered
Properties	-p minSdkVersion:17	Update this property when Encountered
Version	-v 9.9.9	Update the starting repo to this verison
Names	-n "neil simon"	Message in slack of PRs or Error

# Running Hestia

Input Argument	Example	Notes
Commit Message	-m "Updating Support Library"	
JIRA Ticket	-j AND-9999	
Repo Name	-r query-builder	Build up the graph from this repo
Org level change	-o	Update all the repos in the graph
Artifacts	-a com.hootsuite.android:query-builder:1.2.0	Update this artifact when encountered
Properties	-p minSdkVersion:17	Update this property when Encountered
Version	-v 9.9.9	Update the starting repo to this verison
Names	-n "neil simon"	Message in slack of PRs or Error

# Running Hestia

Input Argument	Example	Notes
Commit Message	-m "Updating Support Library"	
JIRA Ticket	-j AND-9999	
Repo Name	-r query-builder	Build up the graph from this repo
Org level change	-o	Update all the repos in the graph
Artifacts	-a com.hootsuite.android:query-builder:1.2.0	Update this artifact when encountered
Properties	-p minSdkVersion:17	Update this property when Encountered
Version	-v 9.9.9	Update the starting repo to this version
Names	-n "neil simon"	Message in slack of PRs or Error

# Running Hestia

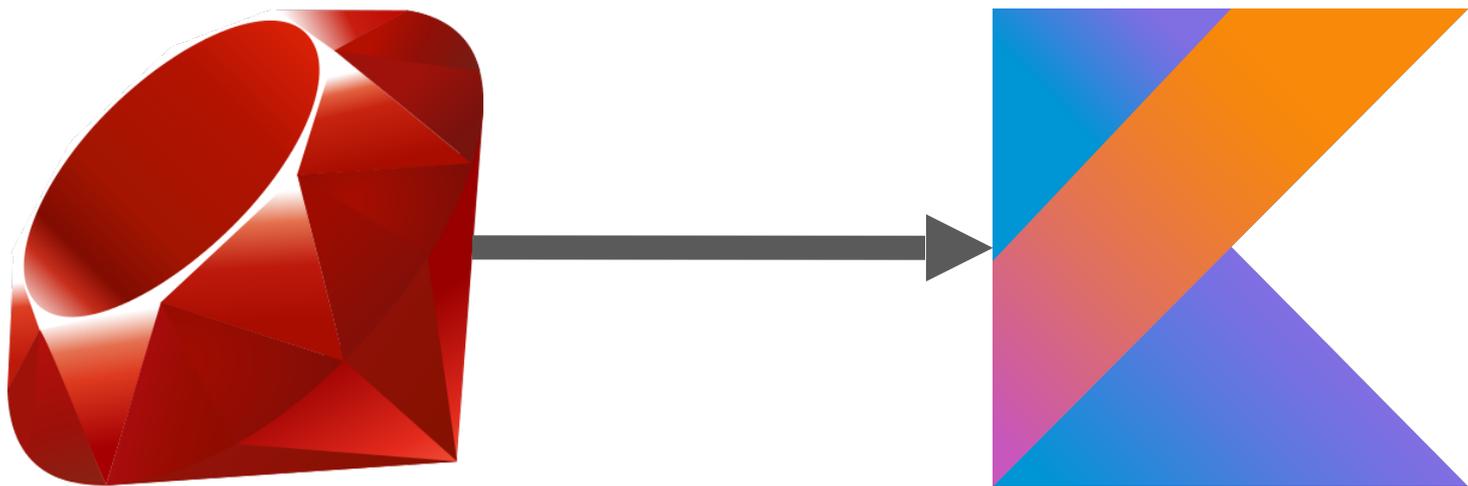
Input Argument	Example	Notes
Commit Message	-m "Updating Support Library"	
JIRA Ticket	-j AND-9999	
Repo Name	-r query-builder	Build up the graph from this repo
Org level change	-o	Update all the repos in the graph
Artifacts	-a com.hootsuite.android:query-builder:1.2.0	Update this artifact when encountered
Properties	-p minSdkVersion:17	Update this property when Encountered
Version	-v 9.9.9	Update the starting repo to this version
Names	-n "neil simon"	Message in slack of PRs or Error

# Working with Kotlin/JS



# Working with Kotlin/JS

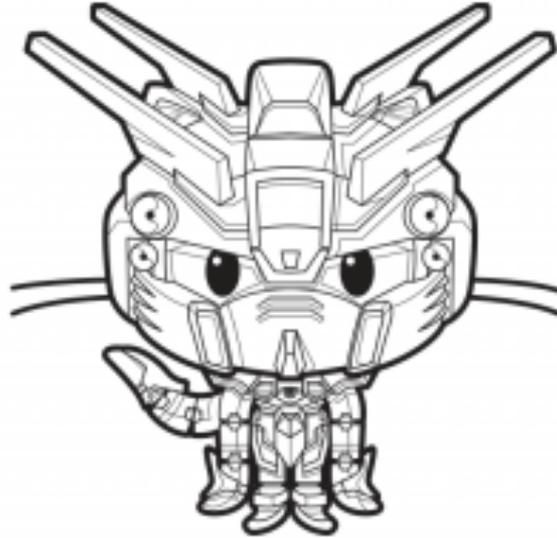
- Both Atlas and Hestia were written in Ruby.



source: <https://www.ruby-lang.org/en/about/logo/>

# Working with Kotlin/JS

- Both implementations use Octokit



source: <https://github.com/octokit>

```
repos = get_client.organization_repositories(organization)
repos.each do |resource|
  repo_list << resource[:name]
end
```



```
begin
  files = get_client.contents(repo, path: path)
  data = nil
  files.each do |file|
    data = Base64.decode64(file[:content])
    PutsUtils.print_with_spacing(fetching_message, 'Success'.colorize(:green))
  end
  return data
rescue Octokit::Error => e
  PutsUtils.print_with_spacing(fetching_message, 'Failure'.colorize(:red))
  puts e
end
end
```



```
external class Repos {
    fun getForOrg(getForOrgConfig: GetForOrgConfig): Promise<GithubArrayResponse<Repo>>
    fun getContent(getContentConfig: GetContentConfig): Promise<GithubResponse<File>>
}

data class GetForOrgConfig(
    val org: String,
    val type: String = "all",
    val page: Int = 1,
    val per_page: Int = 30
)

data class GetContentConfig(
    val owner: String,
    val repo: String,
    val path: String
)

class GithubArrayResponse<T>(val data: Array<T>, val meta: GithubMeta)
class GithubResponse<out T>(val data: T, val meta: GithubMeta)

data class Repo(
    val name: String,
    val html_url: String
)

data class File(
    val name: String,
    val content: String
)

data class GithubMeta(val status: String)
```



```
package com.hootsuite.atlas.visualization

import ...

external class SVG2PNGRequest : Promise<ByteArray>

external fun require(module: String): dynamic

class Visualization {
    private val graphViz = com.hootsuite.atlas.visualization.require(JsModules.GRAPHVIZ)
    private val svg2Png = com.hootsuite.atlas.visualization.require(JsModules.SVG2PNG)

    suspend fun render(graph: List<Artifact>): ByteArray {
        val digraph :String = Digraph(graph).create()
        val svgXML :dynamic = graphViz(digraph, GRAPHVIZ_OPTIONS)
        val request: SVG2PNGRequest = svg2Png(svgXML, SVG2PNG_OPTIONS)
        return request.await()
    }

    companion object {
        private const val GRAPHVIZ_OPTIONS = "{ format: \"svg\" }"
        private const val SVG2PNG_OPTIONS = "{ width: 300, height: 400 }"
    }
}
```

# Working with Kotlin/JS

- Instead of using bundler to bring in Ruby libraries like octokit and graphviz, we are able to use npm to bring in similar libraries.
- Kotlin/JS for Atlas and Hestia provides much needed type safety, null handling, all the features you love about Kotlin in a scripting environment.
- After packaging Atlas as a CommonJS module, we can import it into Hestia and call it similarly to Octokit.
- The next scripting project I start, I will definitely choose Kotlin/JS over something like Ruby.



# Tool Extensibility

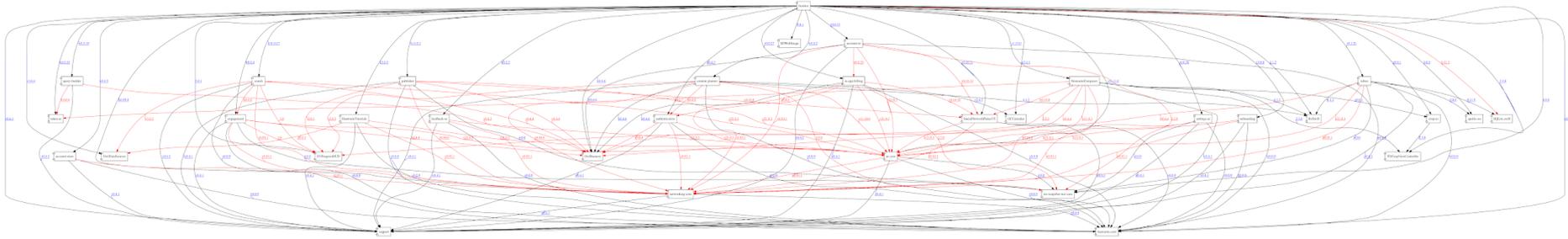


# Atlas

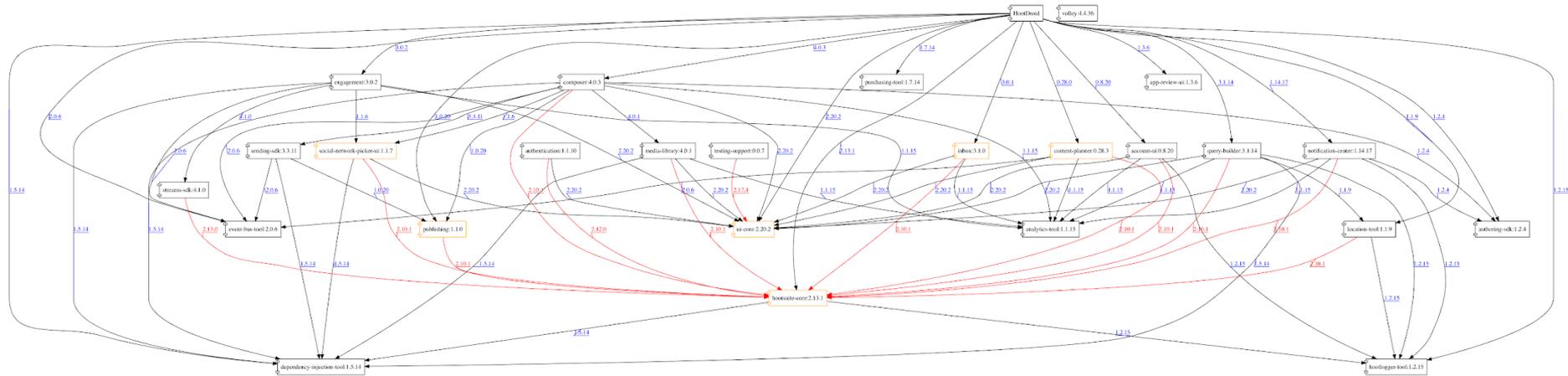
- We use Atlas every day for iOS and Android



# iOS



# Android

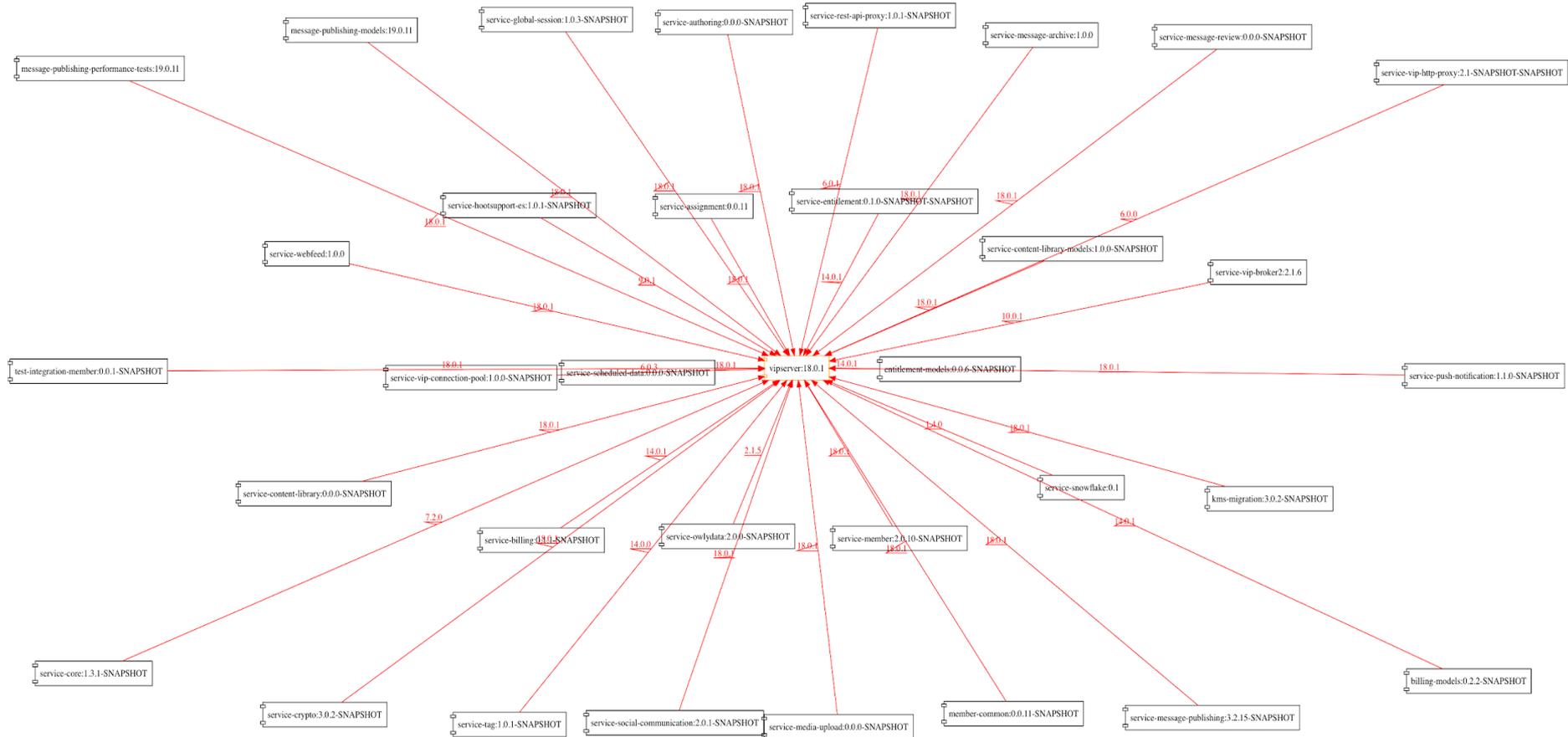


# Scala

- We added SBT support for Atlas to help us visualize our Scala dependencies at Hootsuite.
- Scala at Hootsuite is a much flatter, but much larger graph.
- There are a few key libraries which are depended upon by most Scala services at Hootsuite.



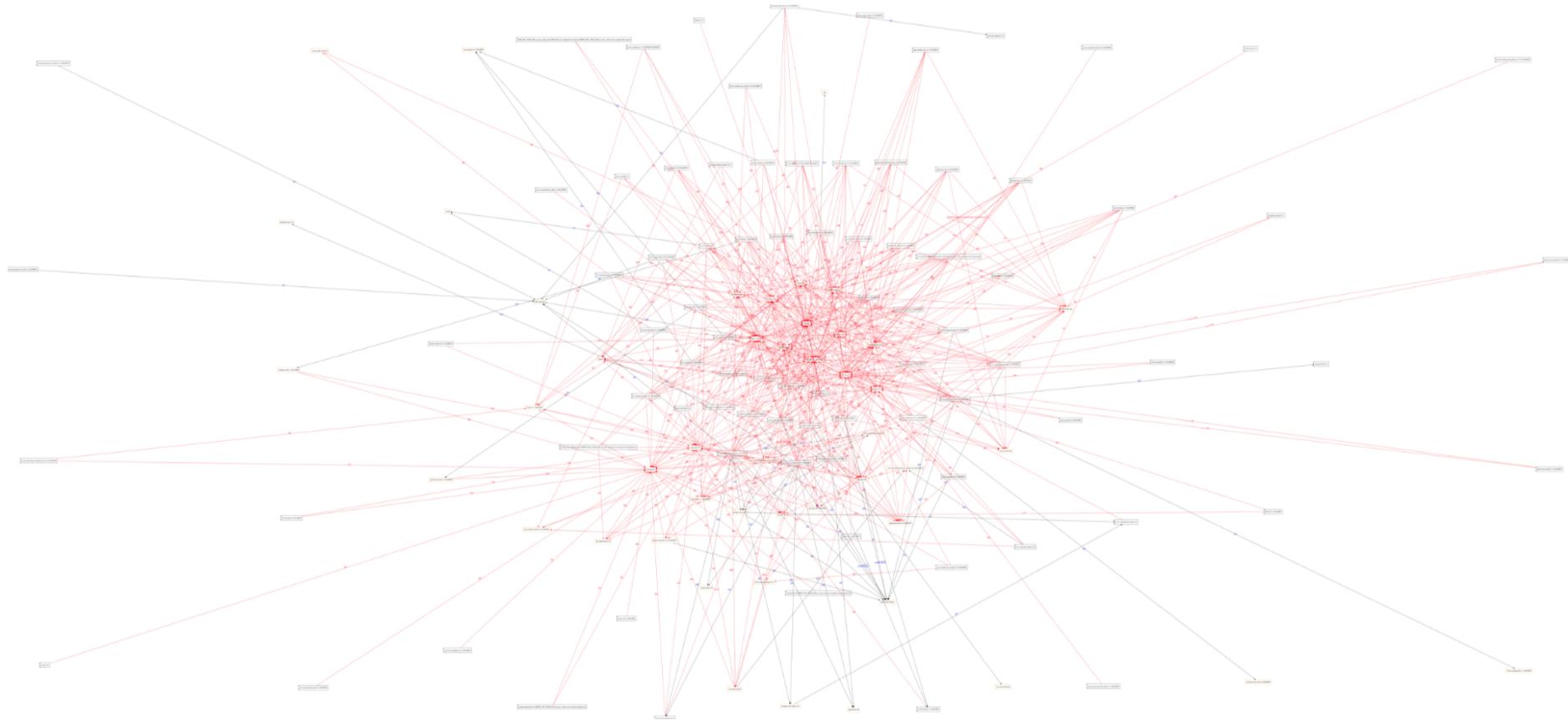
# Hootsuite VIP Server







# Scala at Hootsuite



# 1 - demoing-model-updates - PEON #88

Edit

 **Open** neil-power-hs wants to merge 1 commit into master from peon-demo-1-peon

 Conversation 0    Commits 1    Files changed 1

Changes from all commits ▾ 1 file ▾ +1 -1 ■■■■

Unified Split

Review changes ▾

```
2 ■■■■ project/Version.scala View    
@@ -5,7 +5,7 @@ object Version {  
5 5 // Version of the service models and sdk packages. When changes are made and a new  
6 6 // version is necessary, remove the -SNAPSHOT here, run the build the publish the new version,  
7 7 // then update the minor version by 1 and add -SNAPSHOT, then run the build again.  
8 - val models = "1.0.2-SNAPSHOT"  
8 + val models = "9.9.9"  
9 9  
10 10 // Version of the SDK. Update this whenever the SDK changes. When changes are made and a new  
11 11 // version is necessary, remove the -SNAPSHOT here, run the build the publish the new version,
```



1 - Root Level Change - <https://github.hootops.com/Mobile/service-mobile-api/pull/85>

peon APP 11:31 AM

Job's done: peontest6

1 - Root Level Change - <https://github.hootops.com/Mobile/service-mobile-api/pull/85>



peon APP 11:52 AM ☆

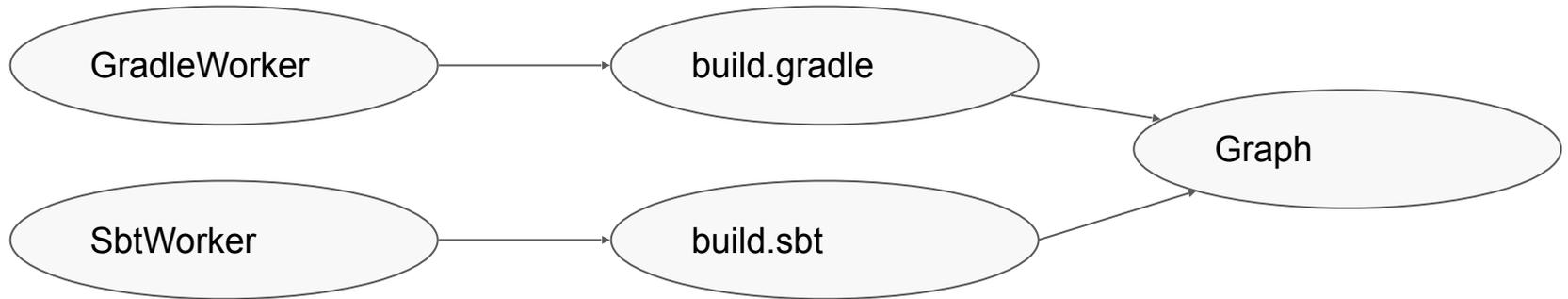
Job's done: peontest7

1 - Root Level Change - <https://github.hootops.com/Mobile/service-mobile-api/pull/86>

Job's done: peontest10

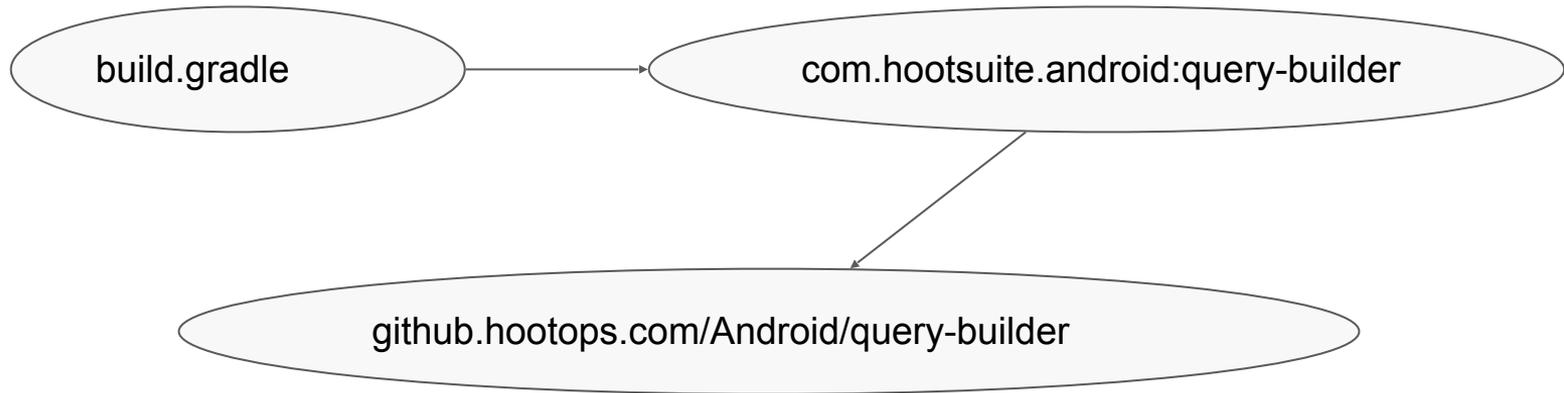
# Atlas Extensibility

- In order to facilitate adding support for other languages and build systems in Atlas, we have created an interface called a “GithubWorker” that we or third party developers can implement.



# Atlas Extensibility

- The worker also needs to know how to find the source code for the dependency. For Hootsuite this is easy, as all our code is on our private Github Enterprise server, but for external dependencies, a whitelist can be used.



# Hestia Extensibility

- After support is added for a language in Atlas, support can be added in Hestia.
- There is a similar interface in Hestia which when implemented, knows how to make alterations to dependency files based on the input I showed above.
- The common aspects of Hestia are reused, syncing repositories, committing changes, creating pull requests, etc.



# Conclusion



# Is It Worth the Time?

HOW LONG CAN YOU WORK ON MAKING A ROUTINE TASK MORE EFFICIENT BEFORE YOU'RE SPENDING MORE TIME THAN YOU SAVE?  
(ACROSS FIVE YEARS)

		HOW OFTEN YOU DO THE TASK					
		50/DAY	5/DAY	DAILY	WEEKLY	MONTHLY	YEARLY
HOW MUCH TIME YOU SHAVE OFF	1 SECOND	1 DAY	2 HOURS	30 MINUTES	4 MINUTES	1 MINUTE	5 SECONDS
	5 SECONDS	5 DAYS	12 HOURS	2 HOURS	21 MINUTES	5 MINUTES	25 SECONDS
	30 SECONDS	4 WEEKS	3 DAYS	12 HOURS	2 HOURS	30 MINUTES	2 MINUTES
	1 MINUTE	8 WEEKS	6 DAYS	1 DAY	4 HOURS	1 HOUR	5 MINUTES
	5 MINUTES	9 MONTHS	4 WEEKS	6 DAYS	21 HOURS	5 HOURS	25 MINUTES
	30 MINUTES		6 MONTHS	5 WEEKS	5 DAYS	1 DAY	2 HOURS
	1 HOUR		10 MONTHS	2 MONTHS	10 DAYS	2 DAYS	5 HOURS
	6 HOURS				2 MONTHS	2 WEEKS	1 DAY
1 DAY					8 WEEKS	5 DAYS	

Source: <https://xkcd.com/1205/>



# The Future of Atlas and Hestia

- I have been working to open source the Kotlin/JS versions of Atlas and Hestia.
  - Goal of finishing this initiative by early November.
  - If you are interested in using Atlas or Hestia, please reach out to me [neil.power@hootsuite.com](mailto:neil.power@hootsuite.com).
  - Both projects will appear here: <https://github.com/hootsuite>



# The Future of Atlas and Hestia

- I have been working to open source the Kotlin/JS versions of Atlas and Hestia.
  - Goal of finishing this initiative by early November.
  - If you are interested in using Atlas or Hestia, please reach out to me [neil.power@hootsuite.com](mailto:neil.power@hootsuite.com).
  - Both projects will appear here: <https://github.com/hootsuite>
- After being Open sourced, our plan is to distribute both tools as CommonJS modules and distribute them through npm.



# The Future of Atlas and Hestia

- I have been working to open source the Kotlin/JS versions of Atlas and Hestia.
  - Goal of finishing this initiative by early November.
  - If you are interested in using Atlas or Hestia, please reach out to me [neil.power@hootsuite.com](mailto:neil.power@hootsuite.com).
  - Both projects will appear here: <https://github.com/hootsuite>
- After being Open sourced, our plan is to distribute both tools as CommonJS modules and distribute them through npm.
- we will continue to work on improving Atlas for SBT and Hestia for SBT.



# The Future of Atlas and Hestia

- I have been working to open source the Kotlin/JS versions of Atlas and Hestia.
  - Goal of finishing this initiative by early November.
  - If you are interested in using Atlas or Hestia, please reach out to me [neil.power@hootsuite.com](mailto:neil.power@hootsuite.com).
  - Both projects will appear here: <https://github.com/hootsuite>
- After being Open sourced, our plan is to distribute both tools as CommonJS modules and distribute them through npm.
- Continue to work on improving Atlas for SBT and Hestia for SBT.
- Allow third parties to implement their own workers so they can use Atlas with their own technologies.



# Other Kotlin Experiments at Hootsuite

- If you want to see another open source Kotlin project, a configurable Github/Slack integration using Ktor, see here: <https://github.com/hootsuite/hermes>



# Shout outs

- Ben Hergert, the original author of Atlas.
- Simon Tse who built Hestia with me the week between Christmas and New Years
- Jim Riecken, Adam Arsenault, Diego Alvarez, Sherry Yuan for helping add Scala support to Atlas
- Ben Hergert and Andy Fisher for helping me rewrite Atlas in Kotlin/JS.



# Thank you! Questions?

**Neil Power**

Senior Software Developer

