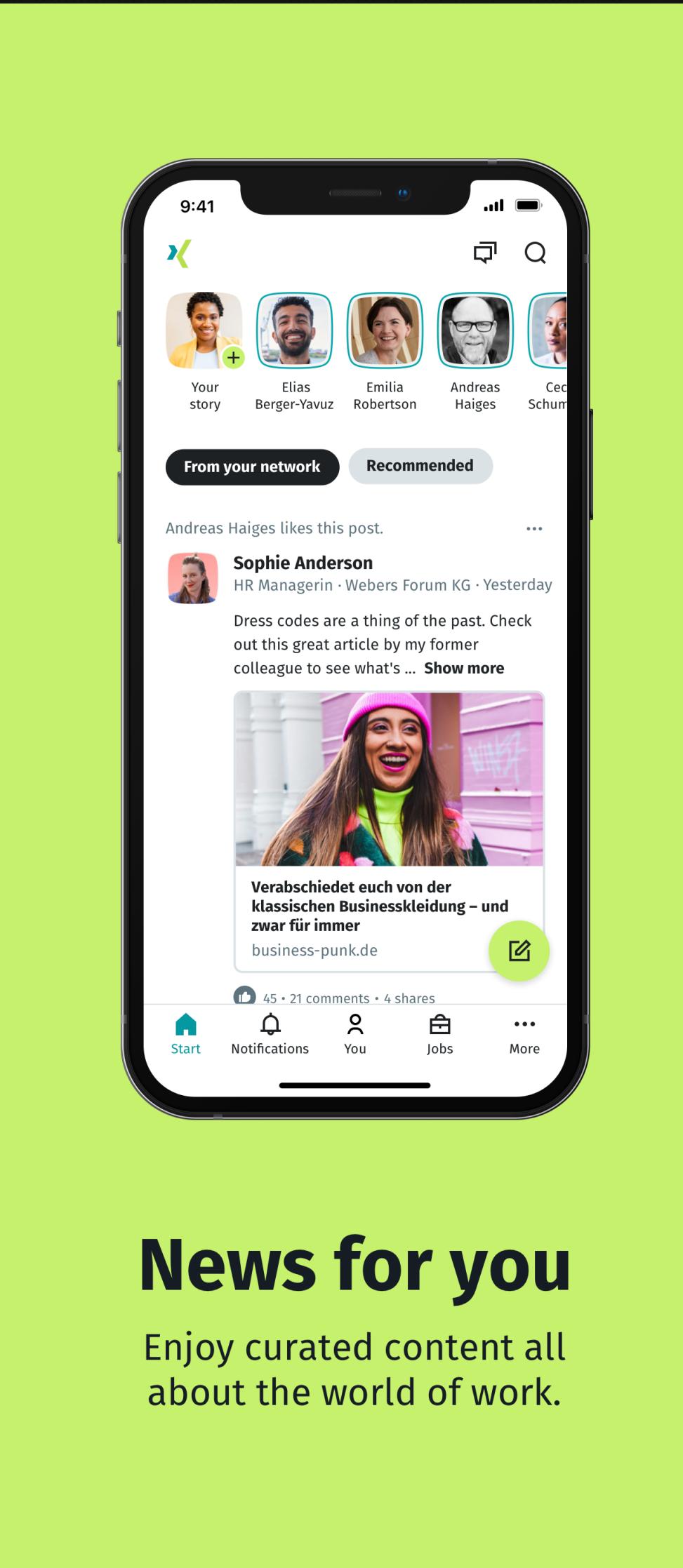


Juantri and Oswaldo

# Playing with Modularization and Dependencies

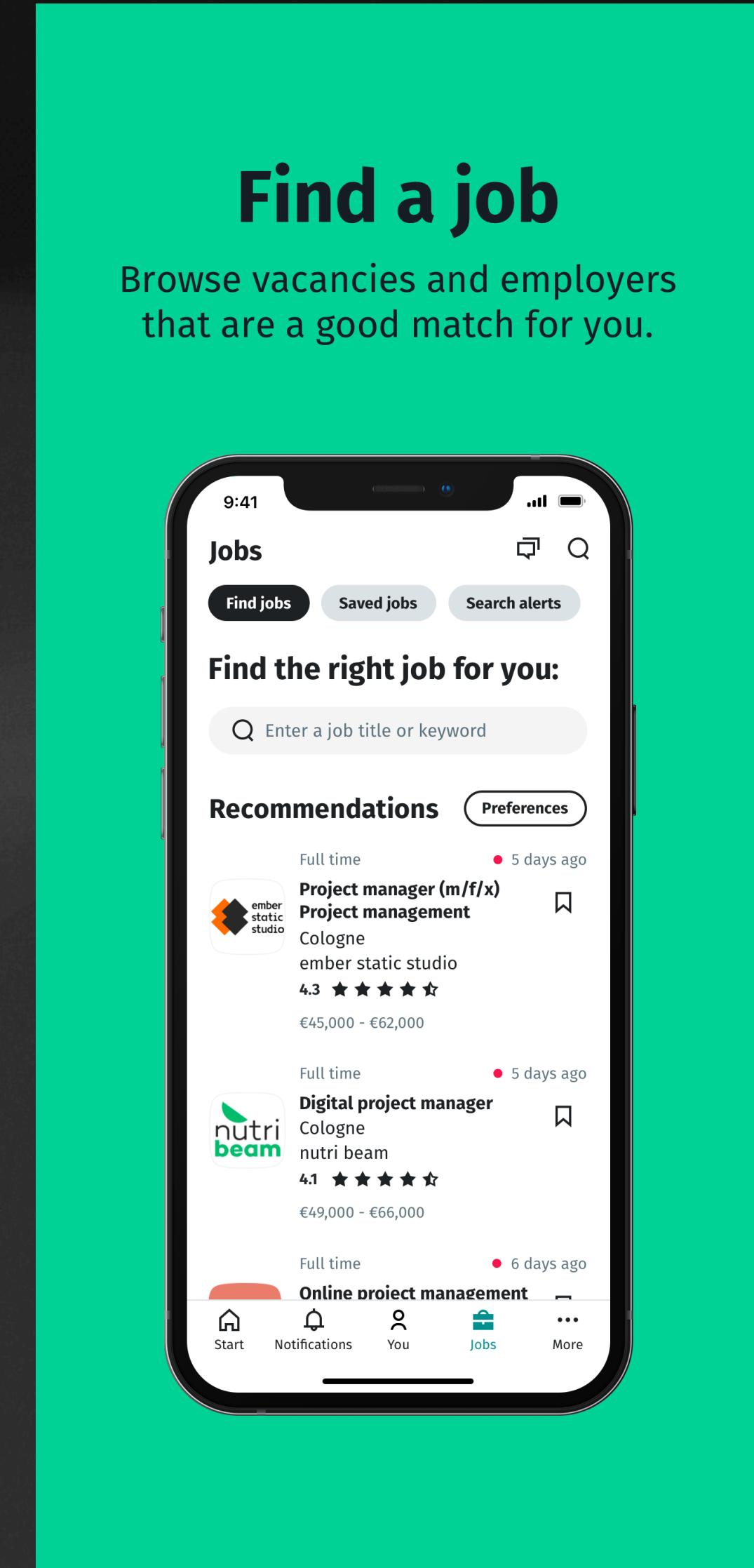
# XING App



The XING app interface on a smartphone screen. At the top, there's a navigation bar with a search icon and a profile icon. Below it, a grid of user profiles shows "Your story" and others like Elias Berger-Yavuz, Emilia Robertson, Andreas Haiges, and Cecilia Schumacher. A post from Sophie Anderson is displayed, showing a photo of a woman in a pink beanie and a caption about告别 classic business clothing. The bottom of the screen features a navigation bar with icons for Start, Notifications, You, Jobs, and More.

**News for you**

Enjoy curated content all about the world of work.



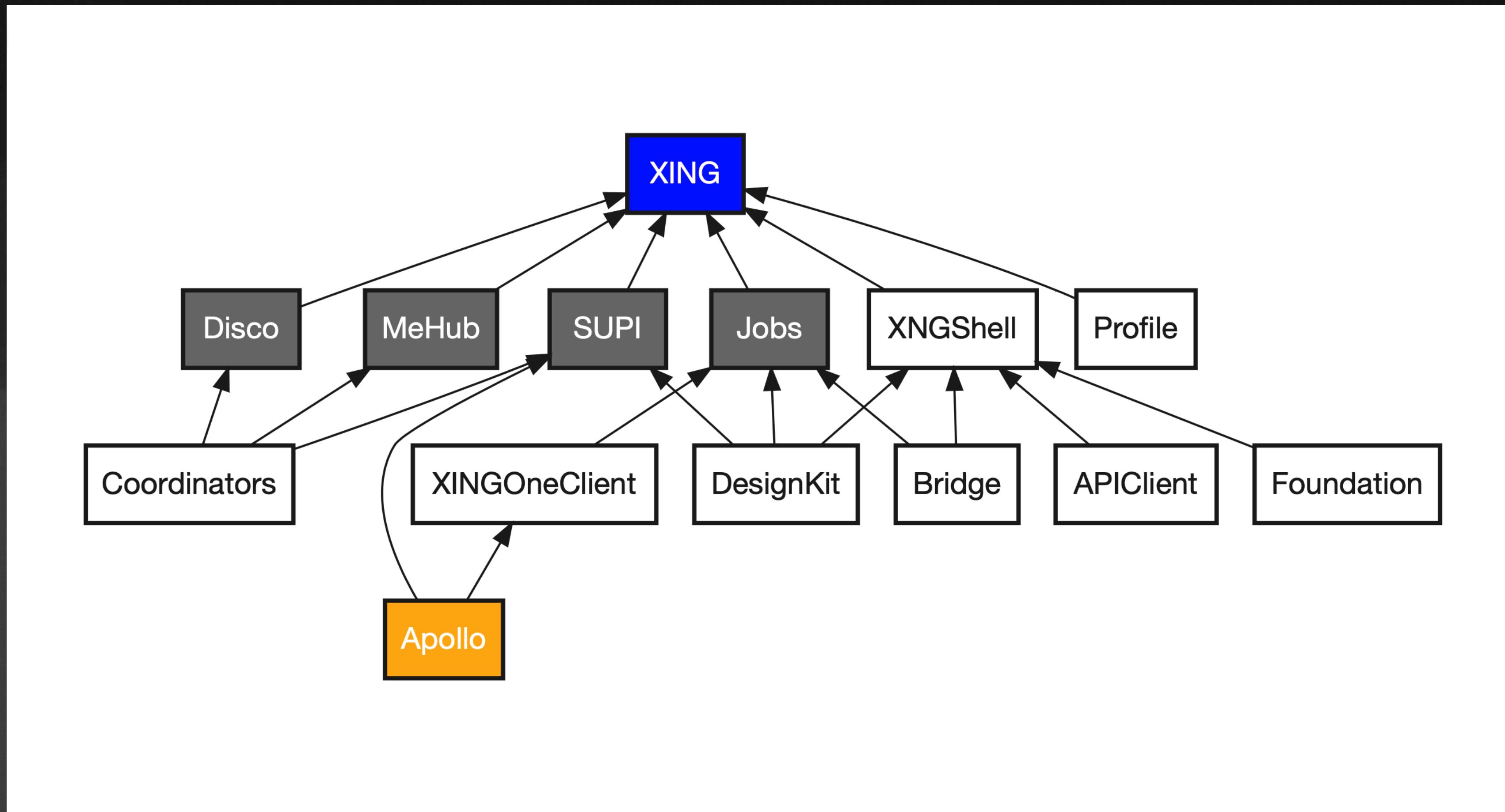
The XING app interface on a smartphone screen. The background is teal. At the top, the text "Find a job" is displayed in large white letters, followed by the subtext "Browse vacancies and employers that are a good match for you." Below this, another smartphone screen shows a job search results page. The title "Find the right job for you:" is at the top, followed by a search bar and a "Recommendations" section. Three job listings are shown:

- Project manager (m/f/x)**  
ember static studio, Cologne  
€45,000 - €62,000
- Digital project manager**  
nutri beam, Cologne  
€49,000 - €66,000
- Online project management**  
ember static studio, Cologne  
€45,000 - €62,000

The bottom of the screen features a navigation bar with icons for Start, Notifications, You, Jobs, and More.



# XING App





# What's the next step?

## Moving to Swift Package Manager

# Current App state

- More than 30 developers spread across 20 teams sharing the same repository
- 110 internal Development Pods with a high dependency between them
- Still a 18% of Objective-C code
- More than 6 years of git repository history and tons of Legacy code

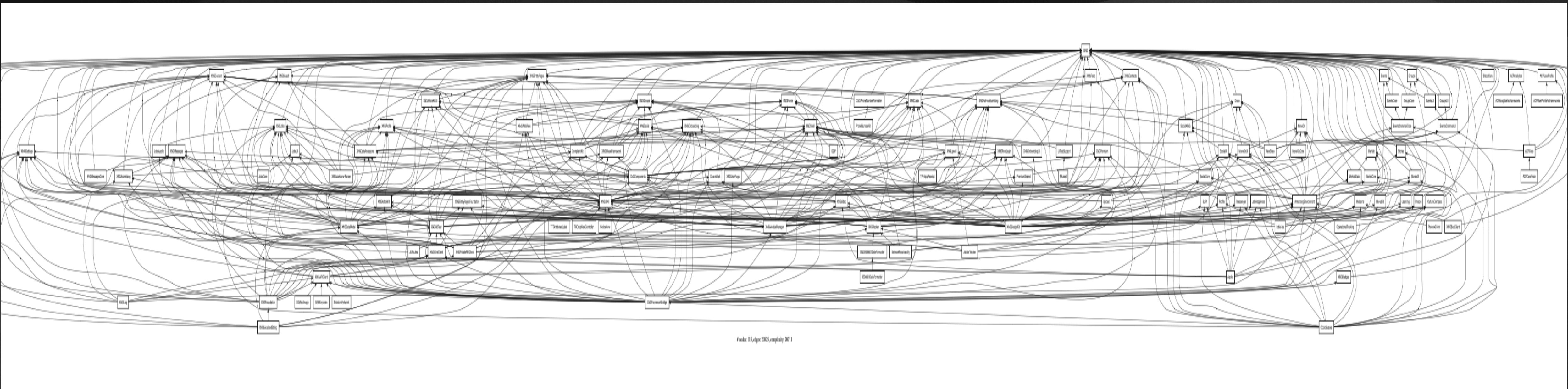
Why do we want to do this now?

- We want to embrace new Swift Package Manager features (native support, easy modularization changes, plugins, build process improvements...)
- We need a more scalable and flexible package manager
- Get rid of all the Cocoapods tooling (ruby, bundler, rvm...)

# How can we do it?



# Our current Dependency Graph for XING app



# Jungle

<https://github.com/xing/jungle>



- Draws your target dependency graph
- Measure the complexity and module number
- All of that along with all your git history
- Compatible with Cocoapods
- Soon also with Swift Package Manager 🤘

$$\text{Complexity} = E - V + P$$

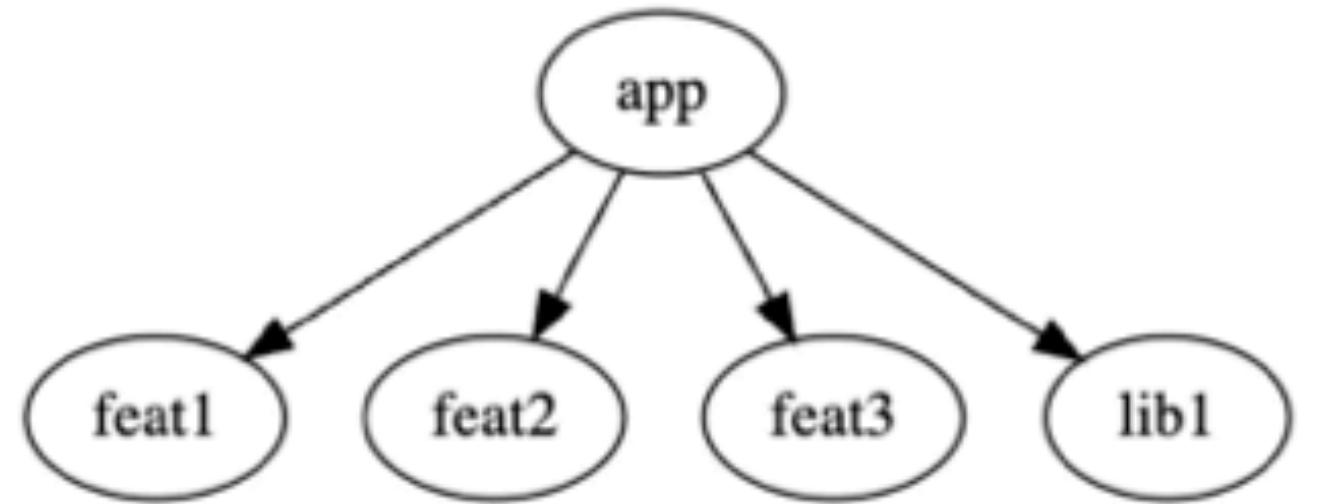
E - Edges

V - Nodes

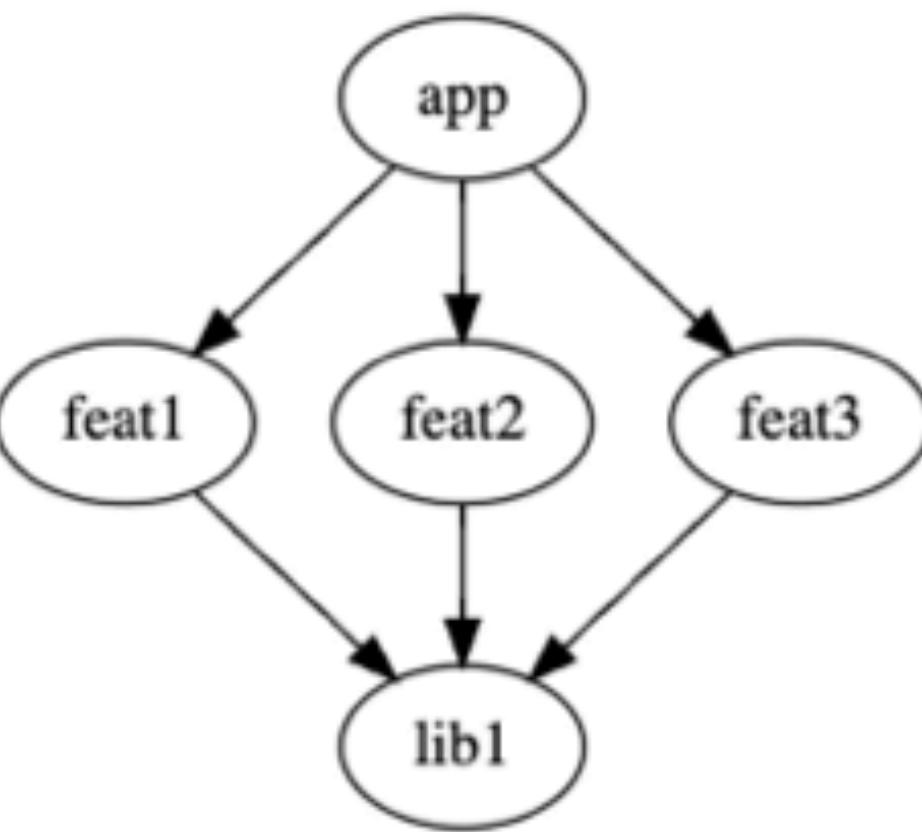
P - Weakly connected components

# Jungle

<https://github.com/xing/jungle>



Complexity:  $4 - 5 + 1 = 0$



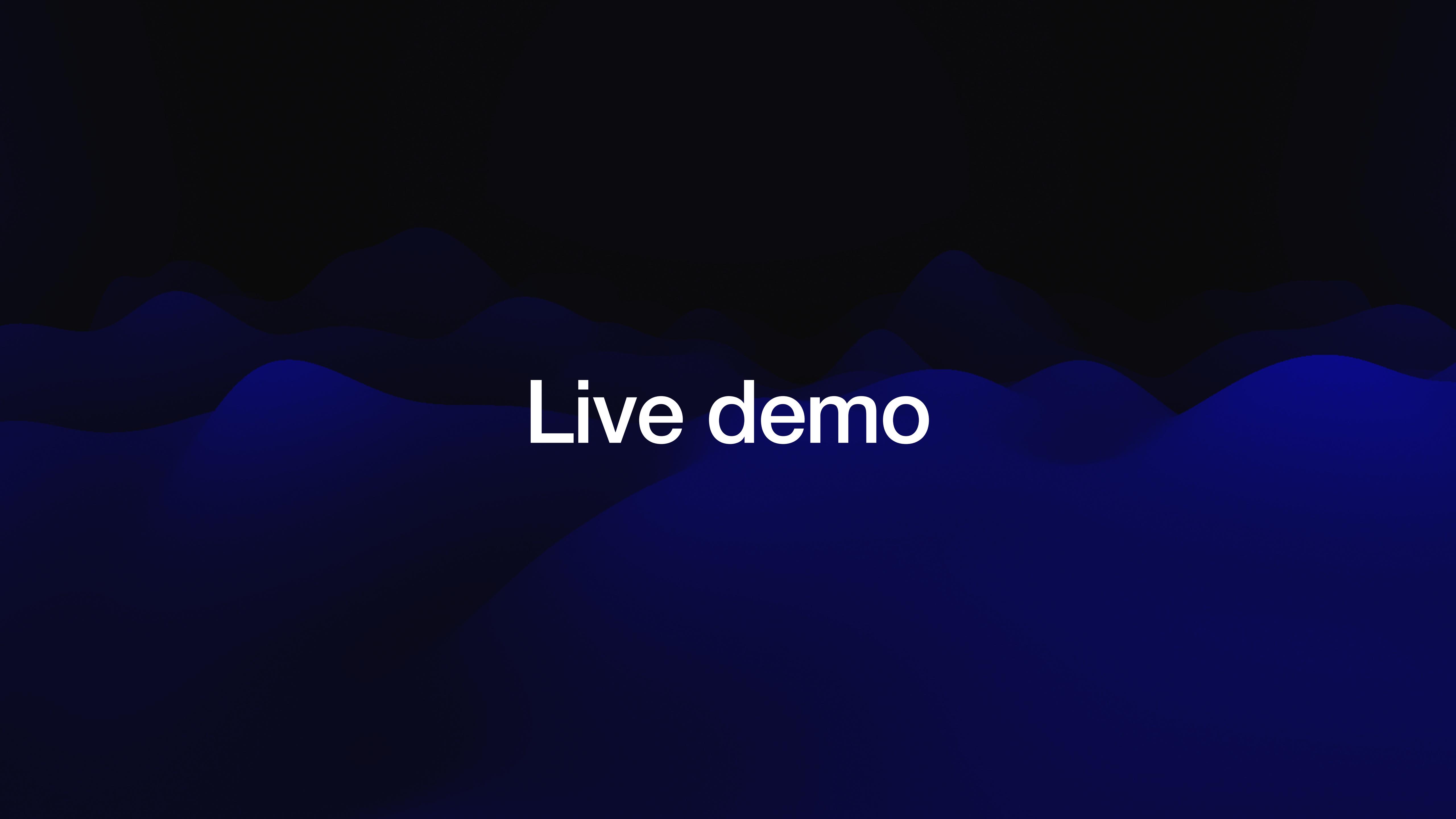
Complexity:  $6 - 5 + 1 = 2$

Complexity =  $E - V + P$

E - Edges

V - Nodes

P - Weakly connected components

The background features a dark blue gradient with three distinct wavy layers. The top layer is a lighter shade of blue, the middle layer is a medium shade, and the bottom layer is a darker shade. These waves create a sense of depth and motion.

Live demo

# Where we should add a new dependency?

Main complexity = 21066

- APIClient
- Bridge
- Foundation
- Feature

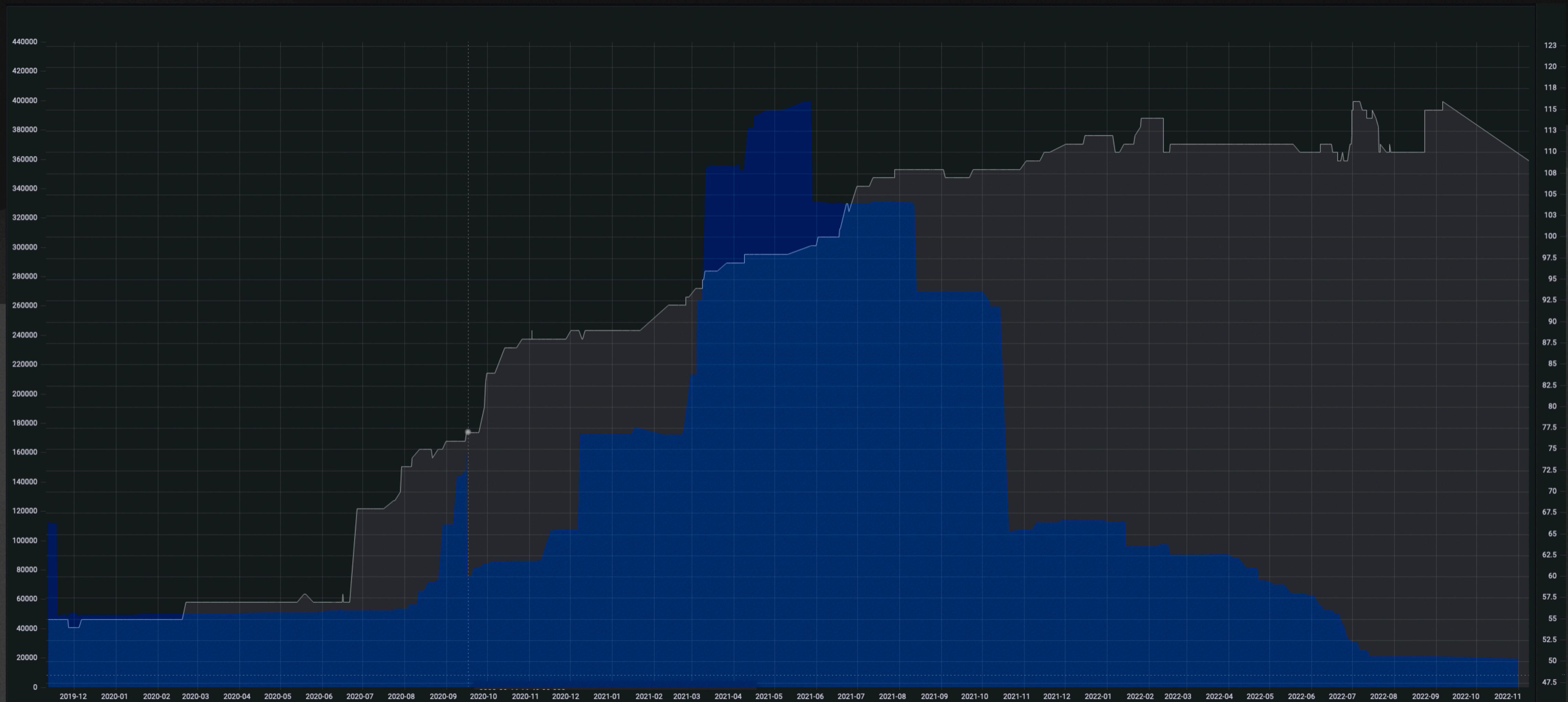
# Where we should add a new dependency?

Main complexity = 21066

- APIClient:  $22026/21066 - 1 = +4.5\%$
- Bridge:  $23360/21066 - 1 = +10.8\%$
- Foundation:  $23022/21066 - 1 = +9.3\%$
- Feature:  $21127/21066 - 1 = +0.3\%$

# Last 3 years of XING iOS project

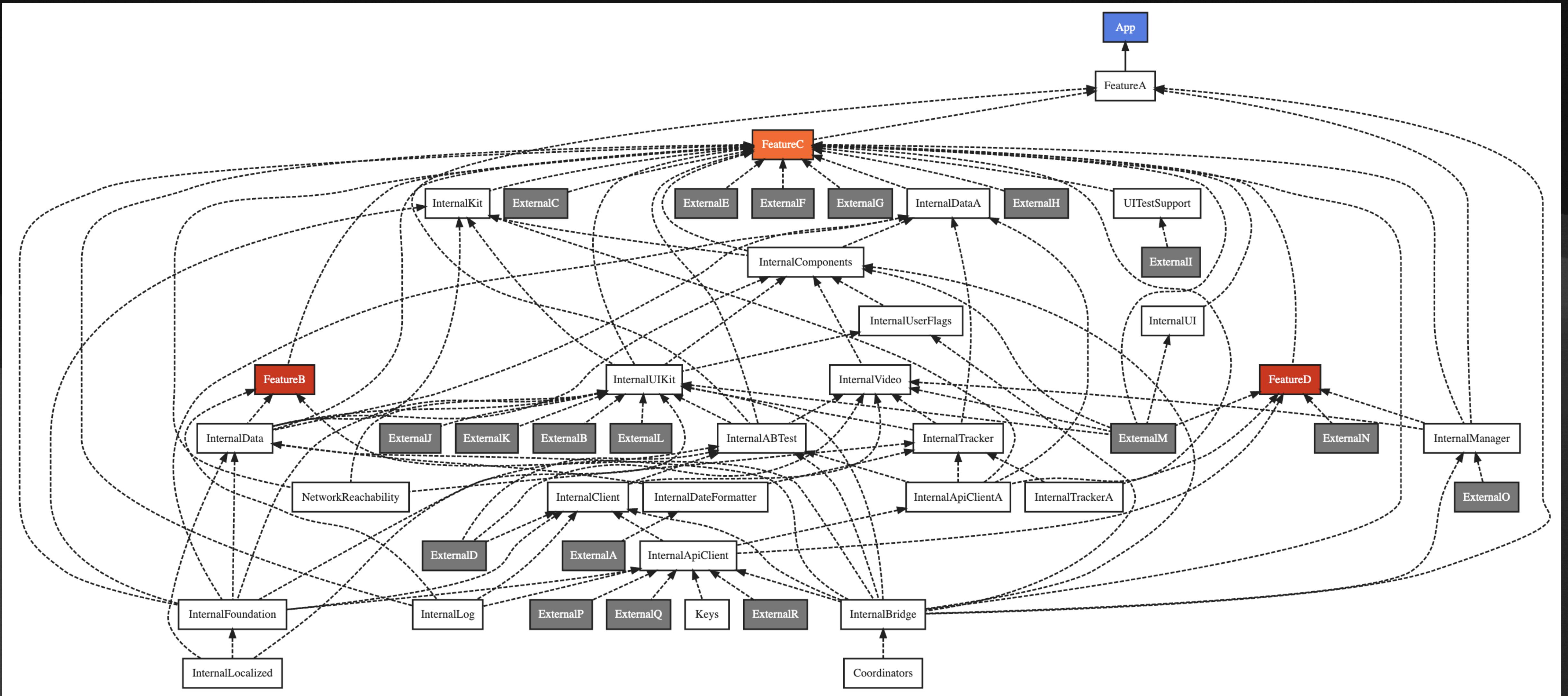
## Complexity and Modules evolution



The background features a dark blue gradient with three distinct wavy layers. The top layer is a very dark navy blue, the middle layer is a medium-dark blue, and the bottom layer is a bright navy blue. These layers create a sense of depth and motion.

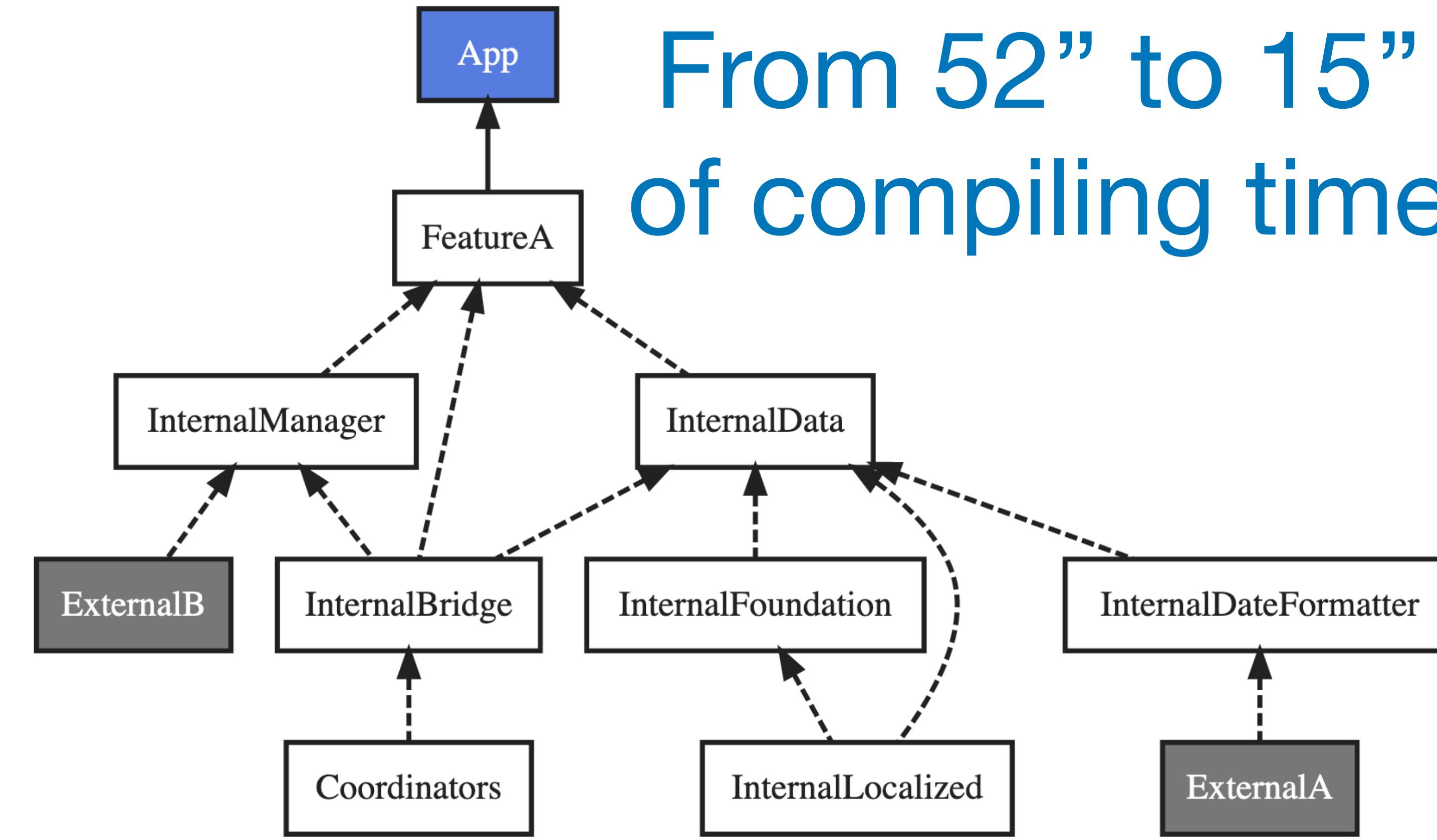
DI is your friend!

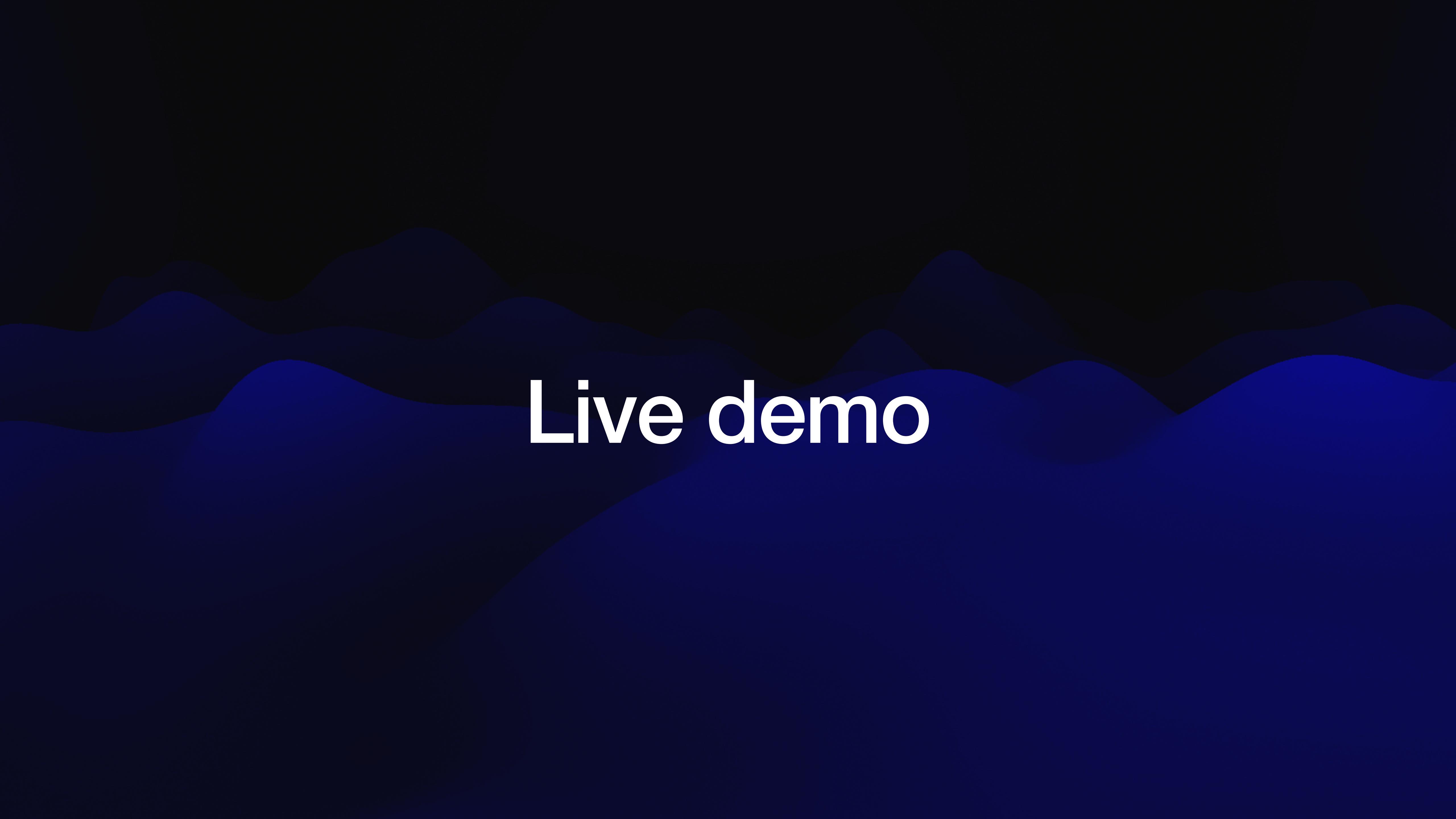
# Applying DI in the XING App



# Applying DI in the XING App

From 52" to 15"  
of compiling time

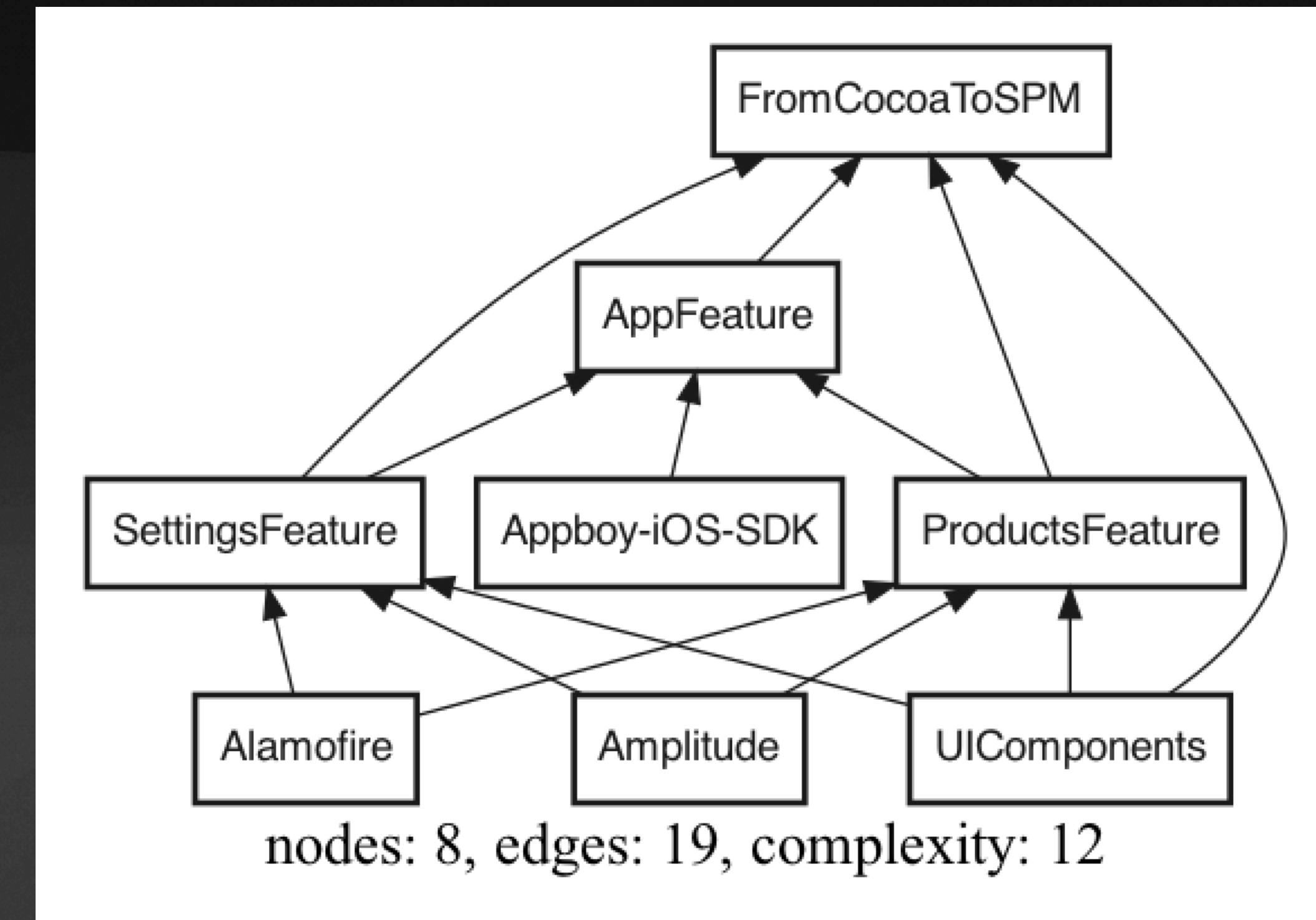


The background features a dark blue gradient with three distinct wavy layers. The top layer is a very dark navy blue, the middle layer is a medium-dark blue, and the bottom layer is a bright navy blue. These layers create a sense of depth and motion.

Live demo

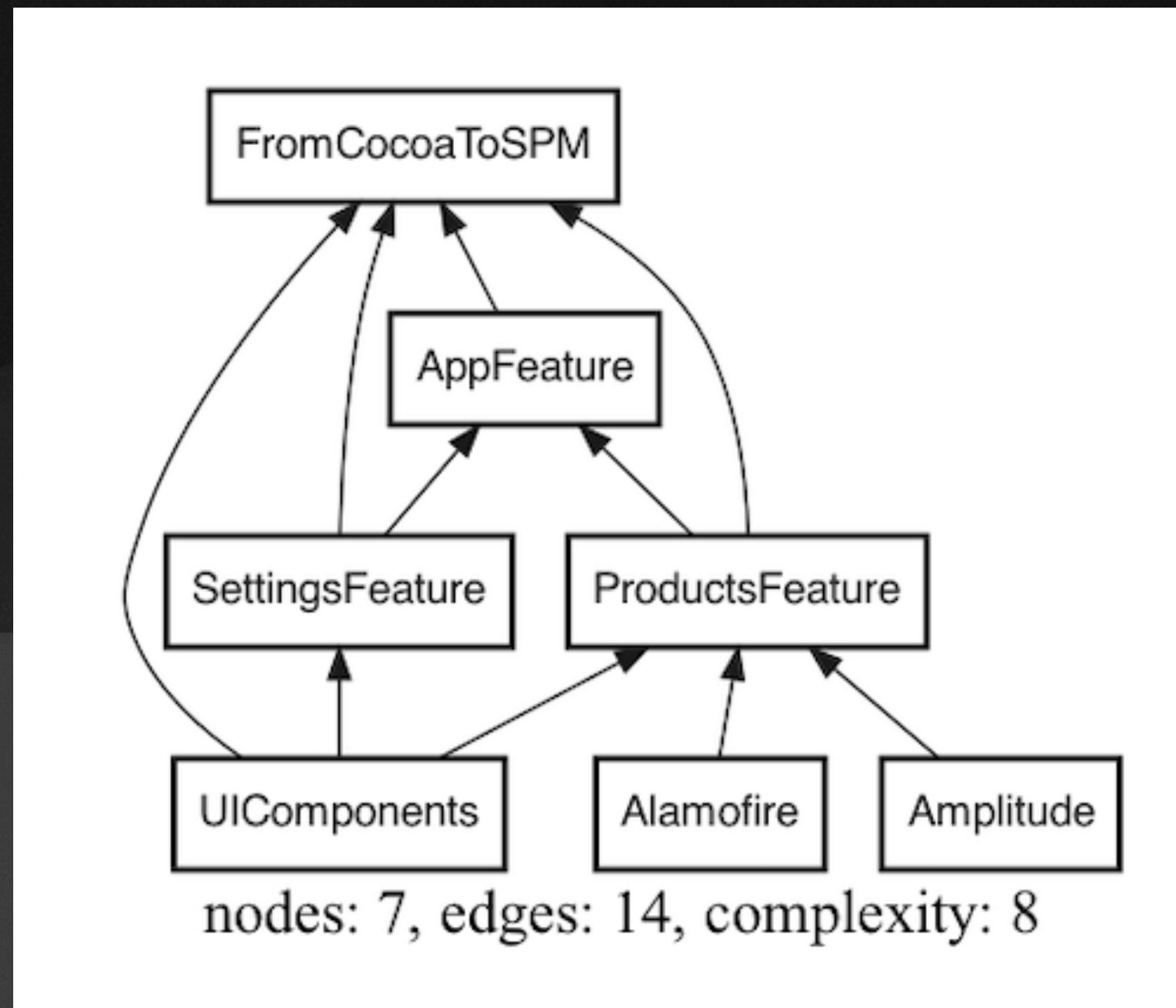
# Step 1 - Measuring the complexity

# Initial graph and complexity

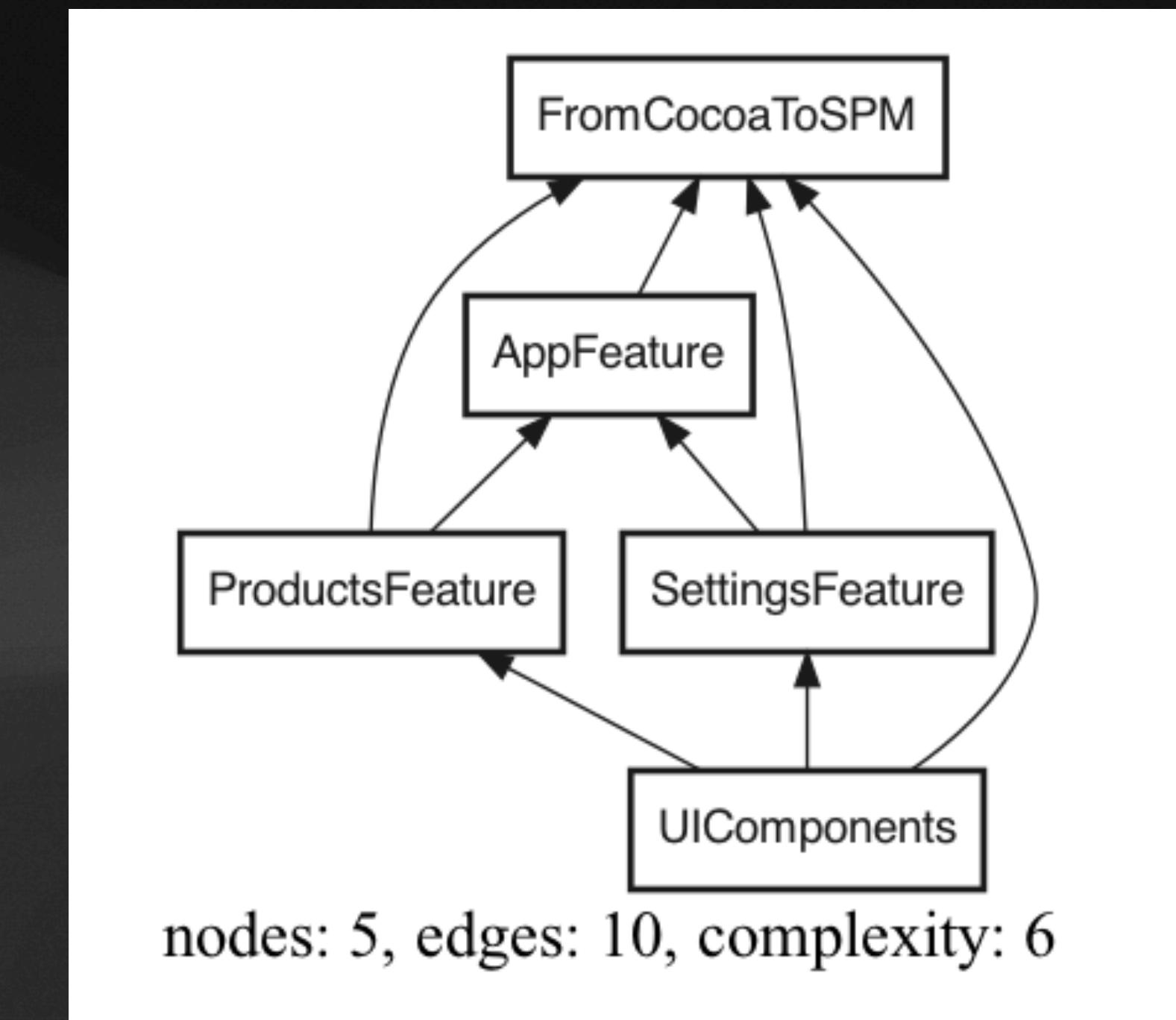
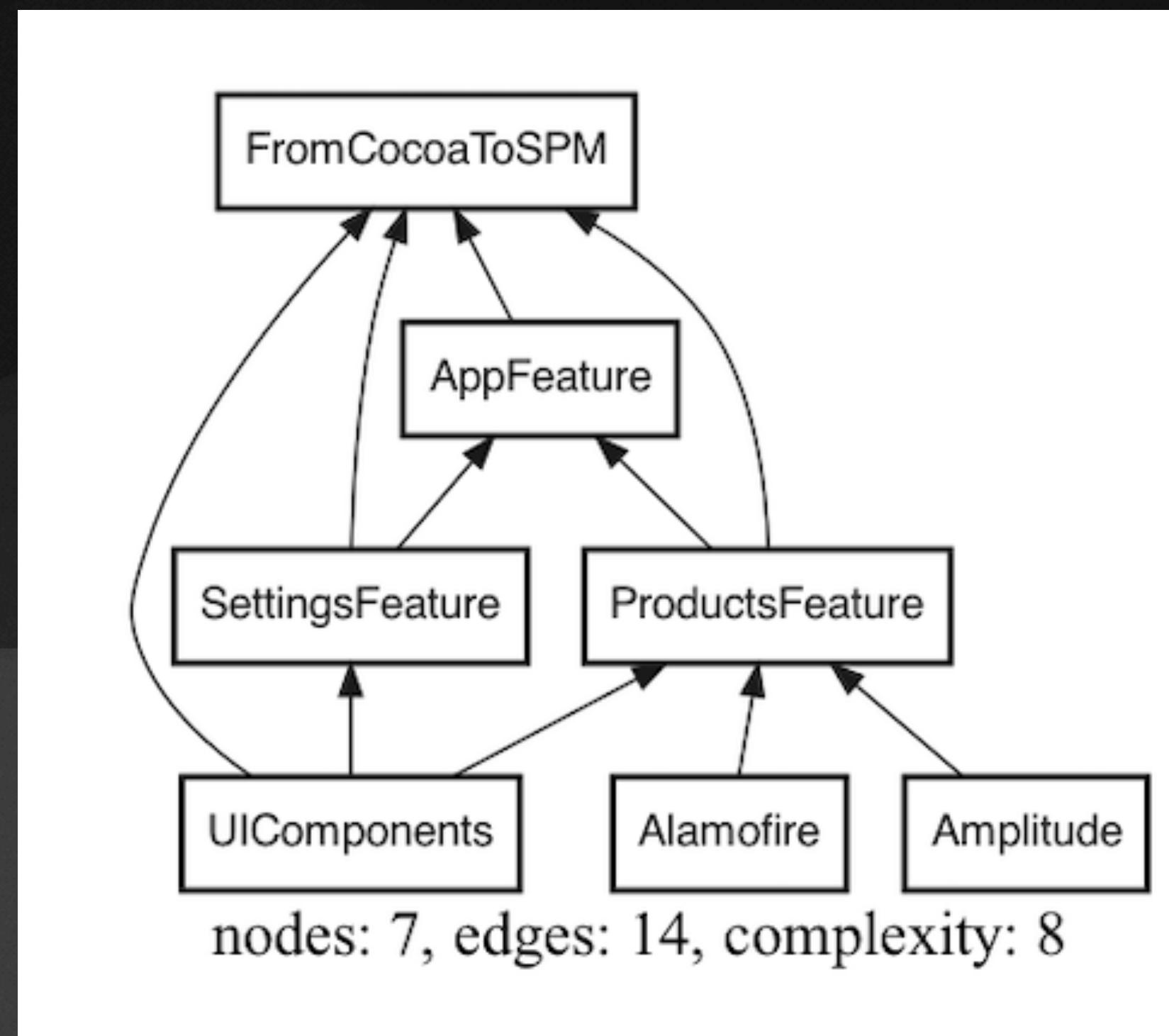


# Step 2 - Reducing the complexity

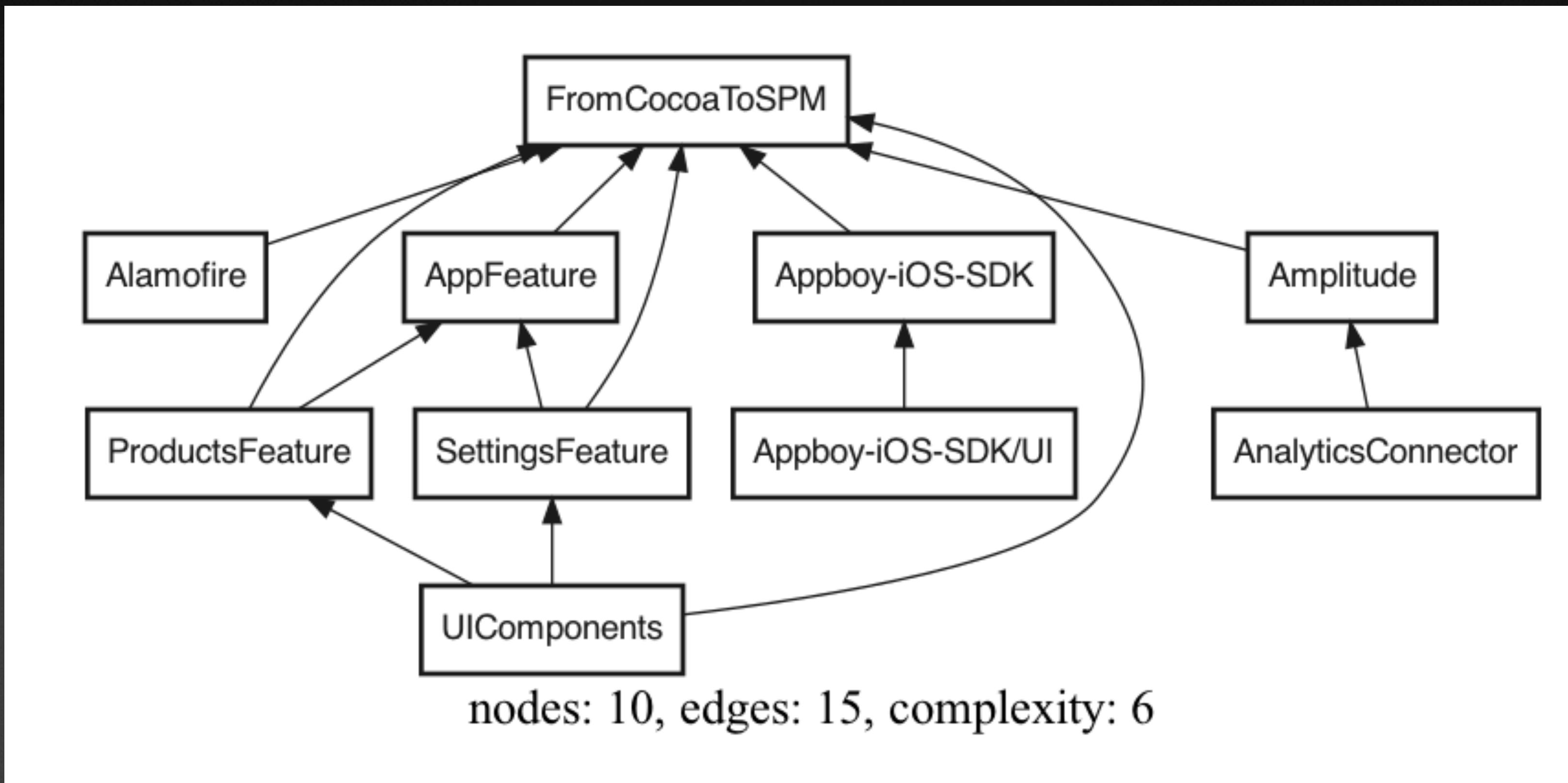
# External dependencies removed from SettingsFeature



# External dependencies removed from ProductsFeature

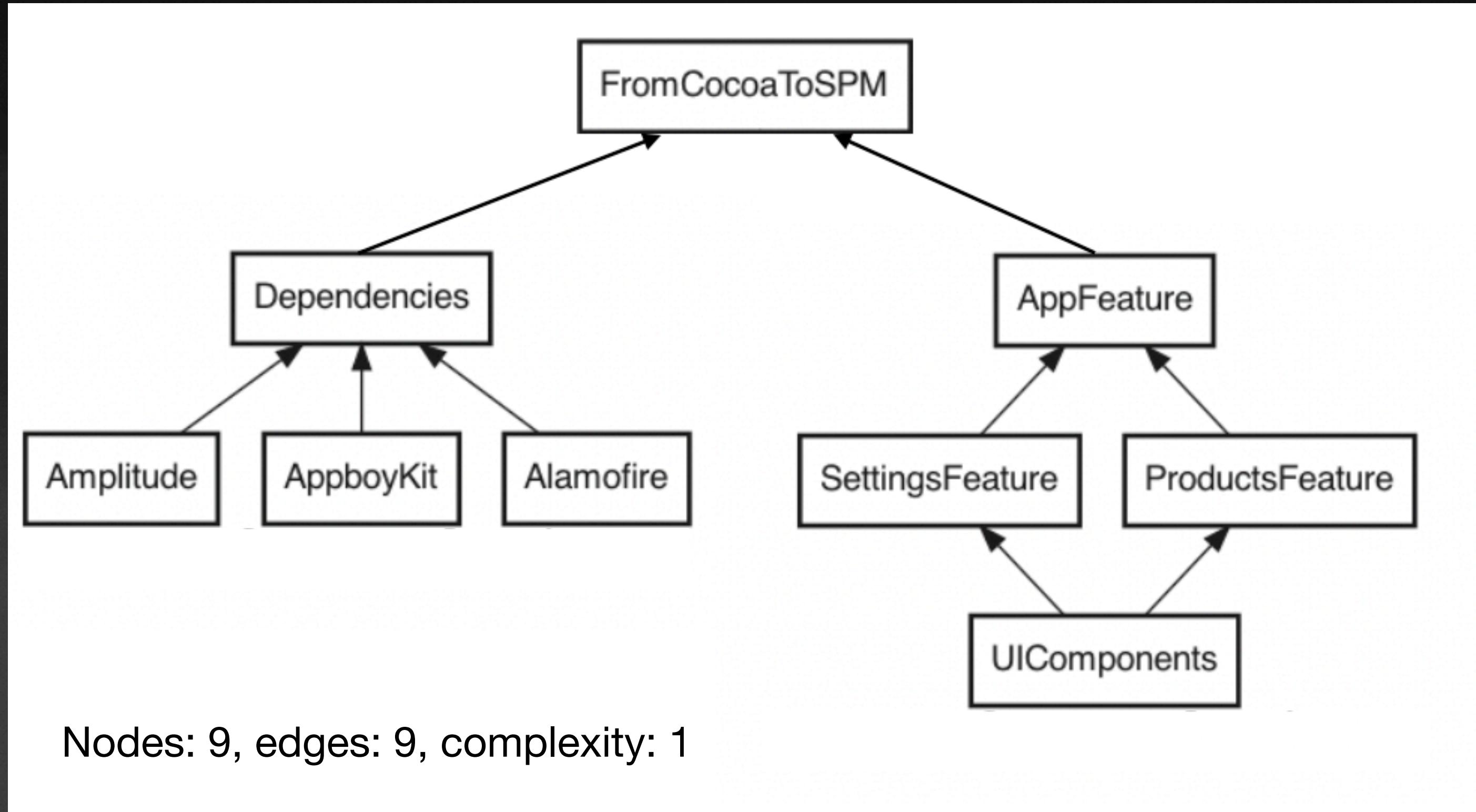


# No external dependencies used by internal ones



# Step 3 - Switching to SPM

# Final dependency graph and complexity



# Conclusions

- Reducing the dependency level between your modules should be always the first step.
- Measure the changes your team is doing on your project and react based on that.
- Keep doing.

# Contact us

- [twitter.com/arrozconnori](https://twitter.com/arrozconnori)
- [twitter.com/juantri94](https://twitter.com/juantri94)

