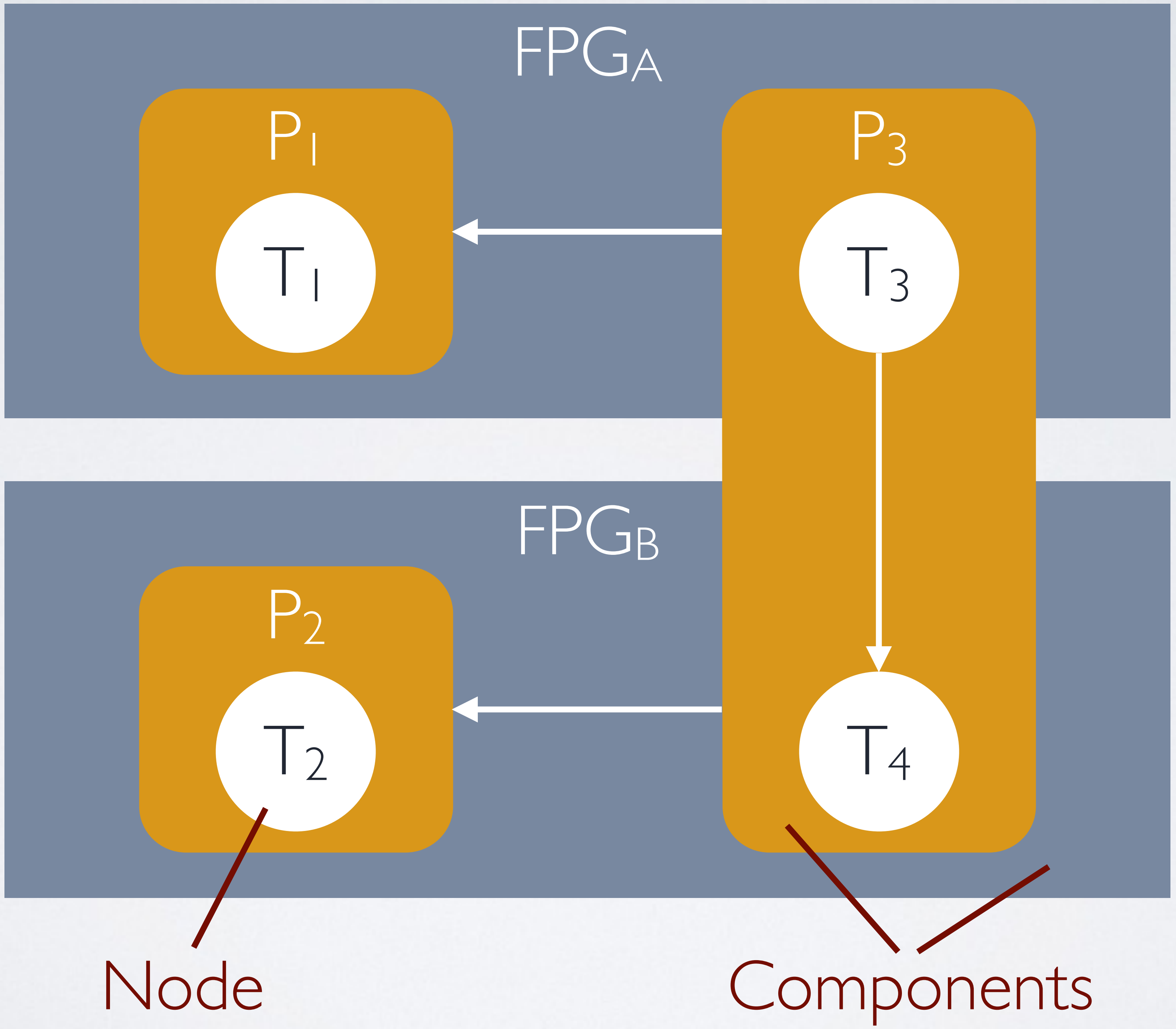
GREETINGS FROM MELBOURNE

…and Stuff!

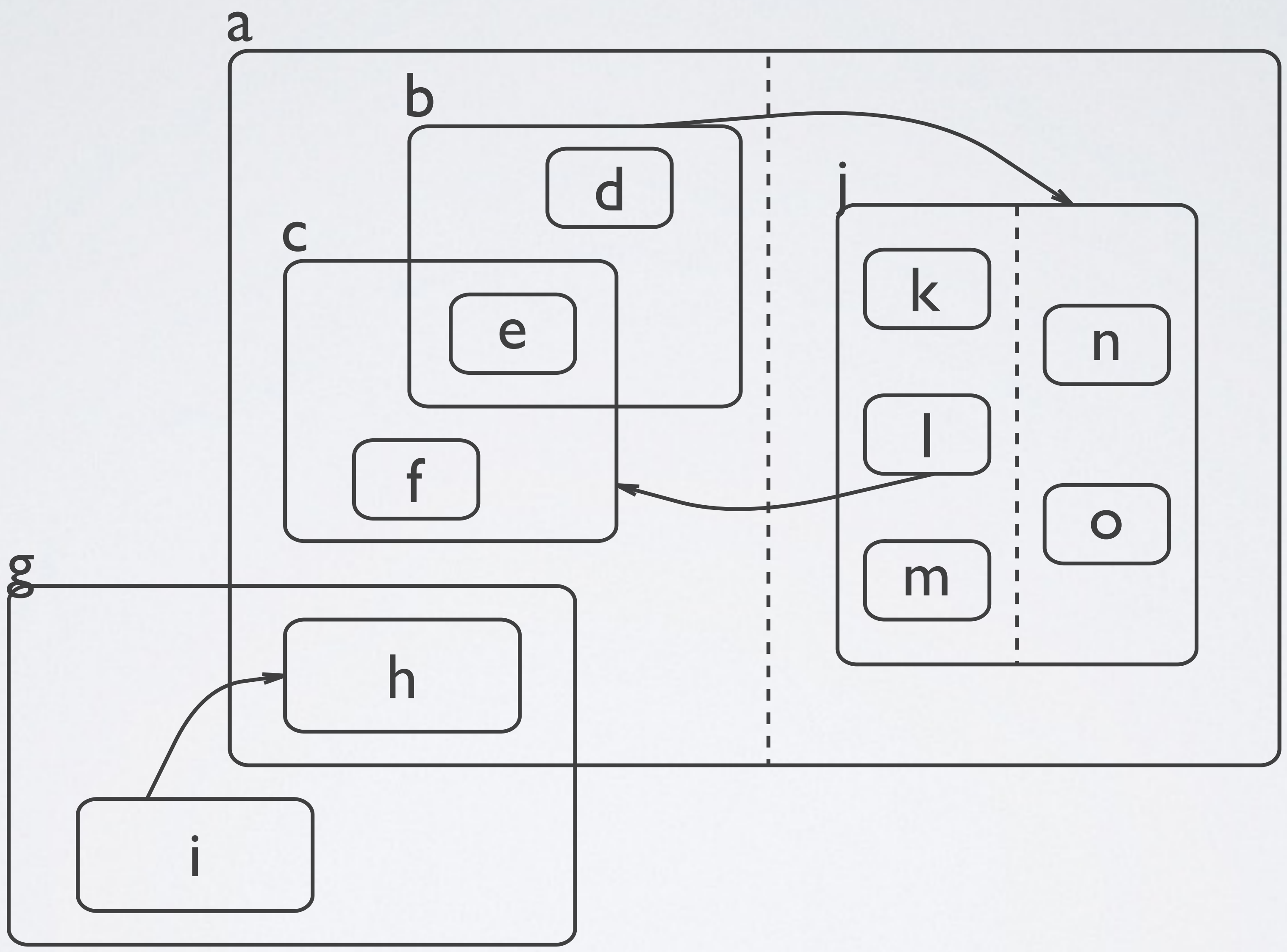
GREETINGS FROM AUSTIN

…and Stuff!

Component Diagrams



Higraphs



Grossman, Ornit, Harel · *On the Algorithmics of Higraphs*
Technical Report CS97-15

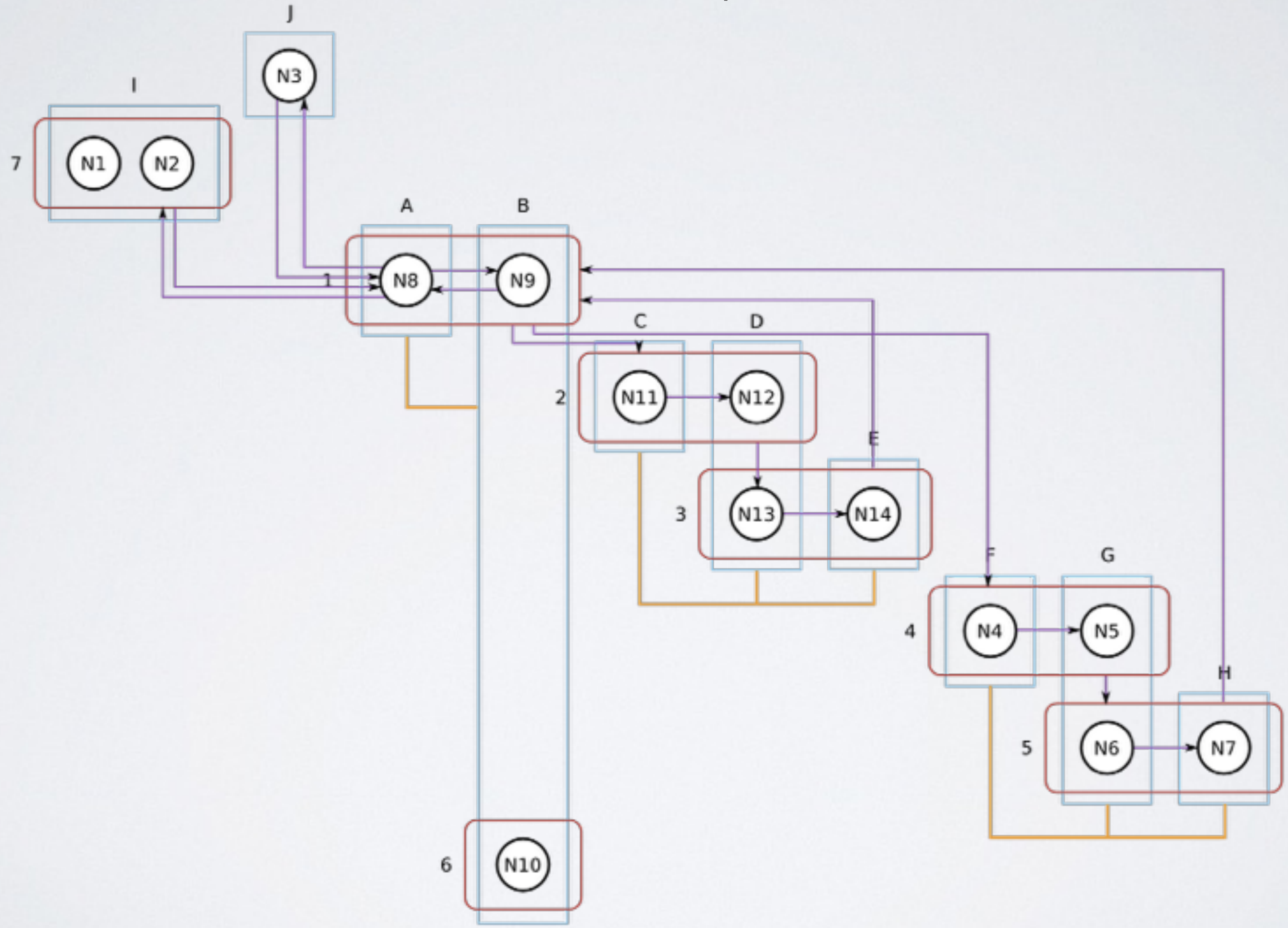
Constraint

At most 2

Component types

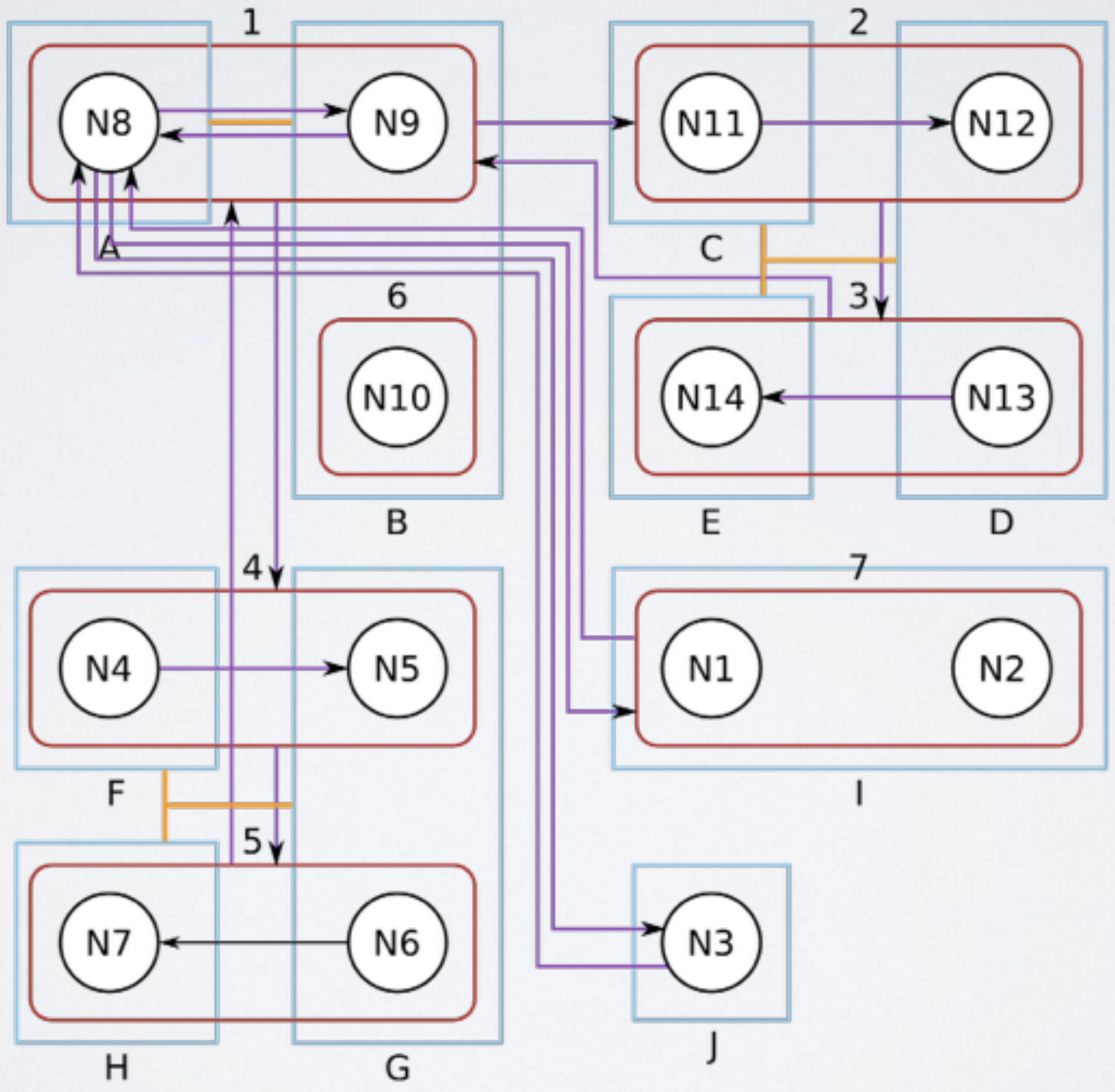
First Ideas

Tabular Layout

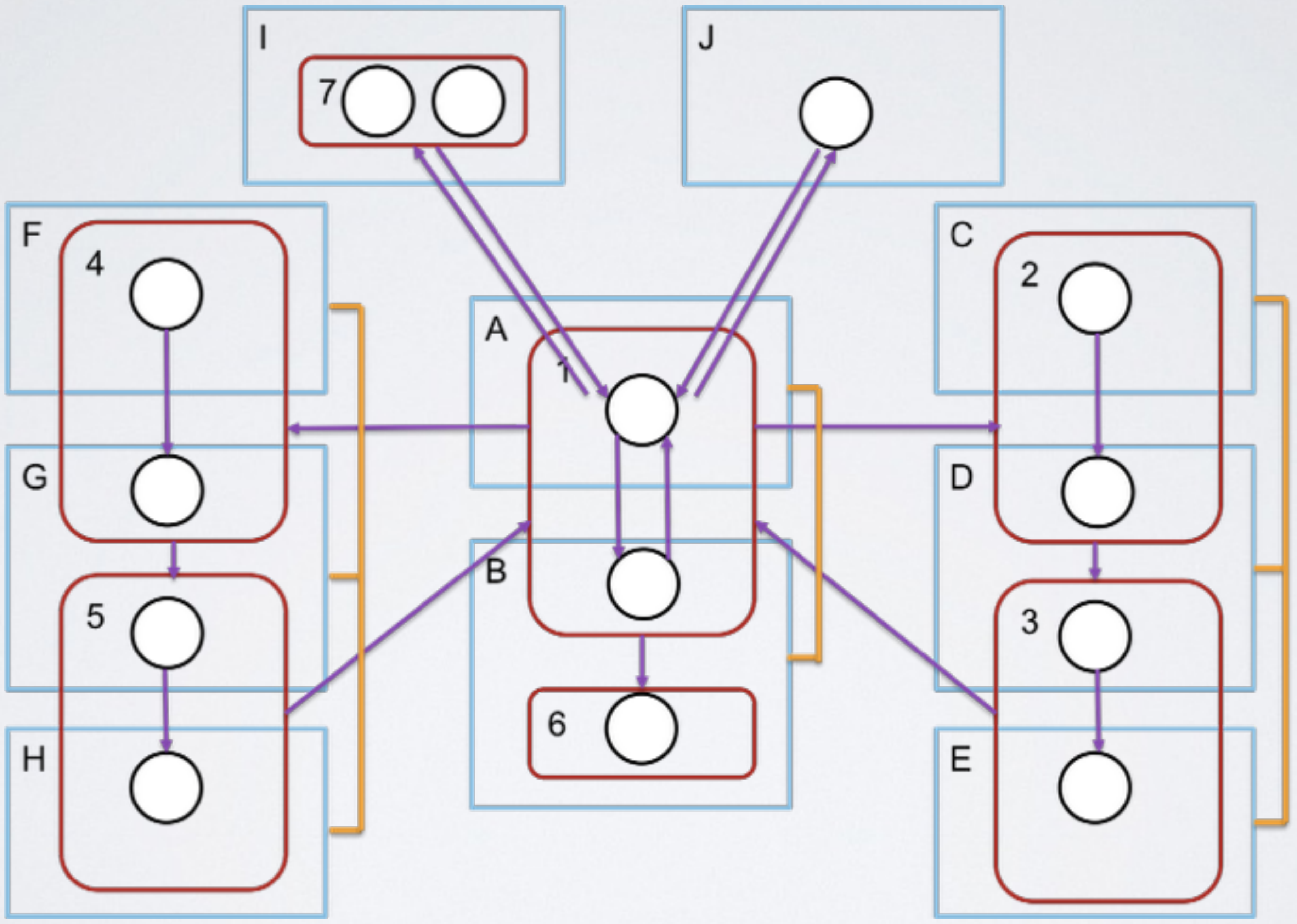


First Ideas

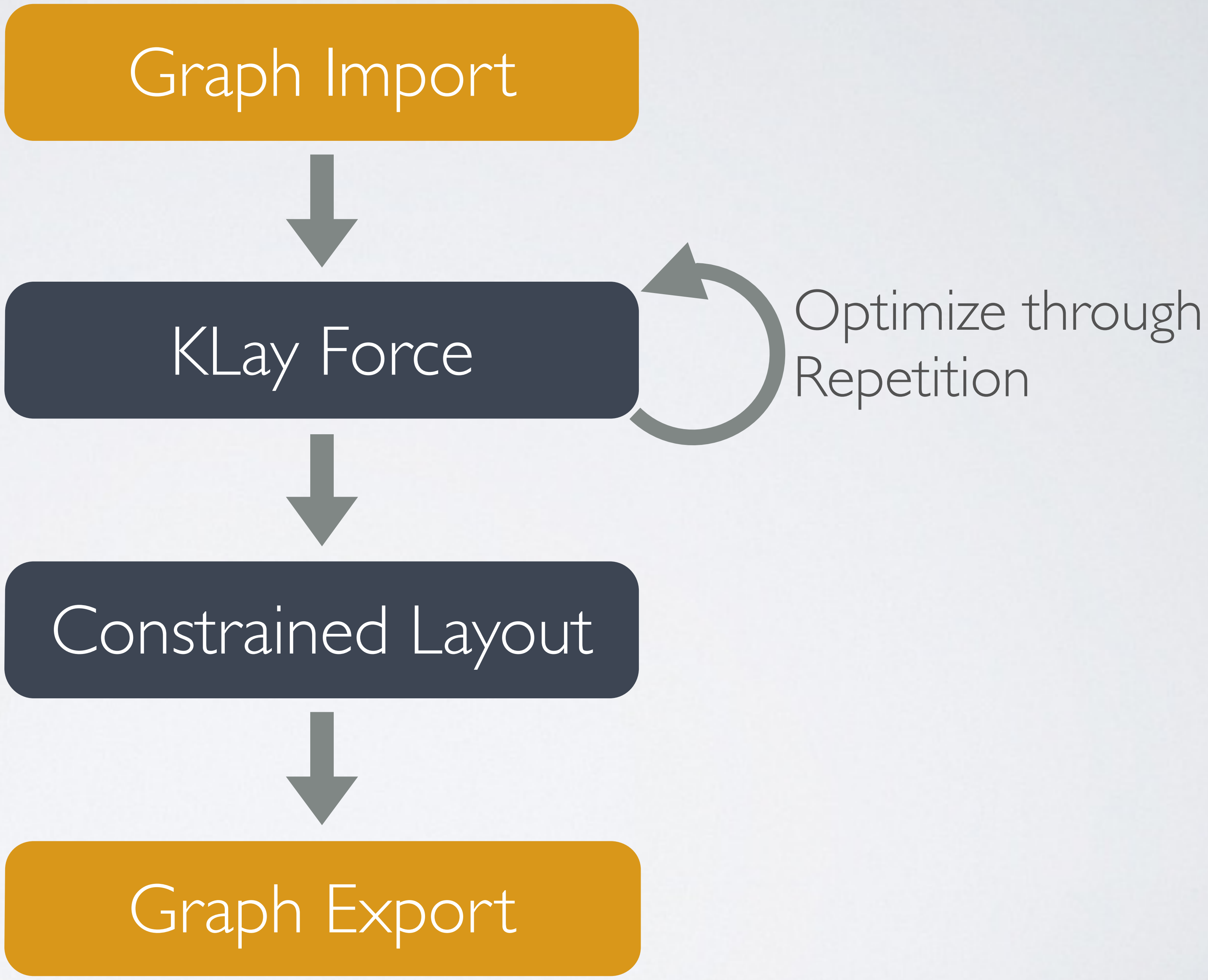
Orthogonal Layout



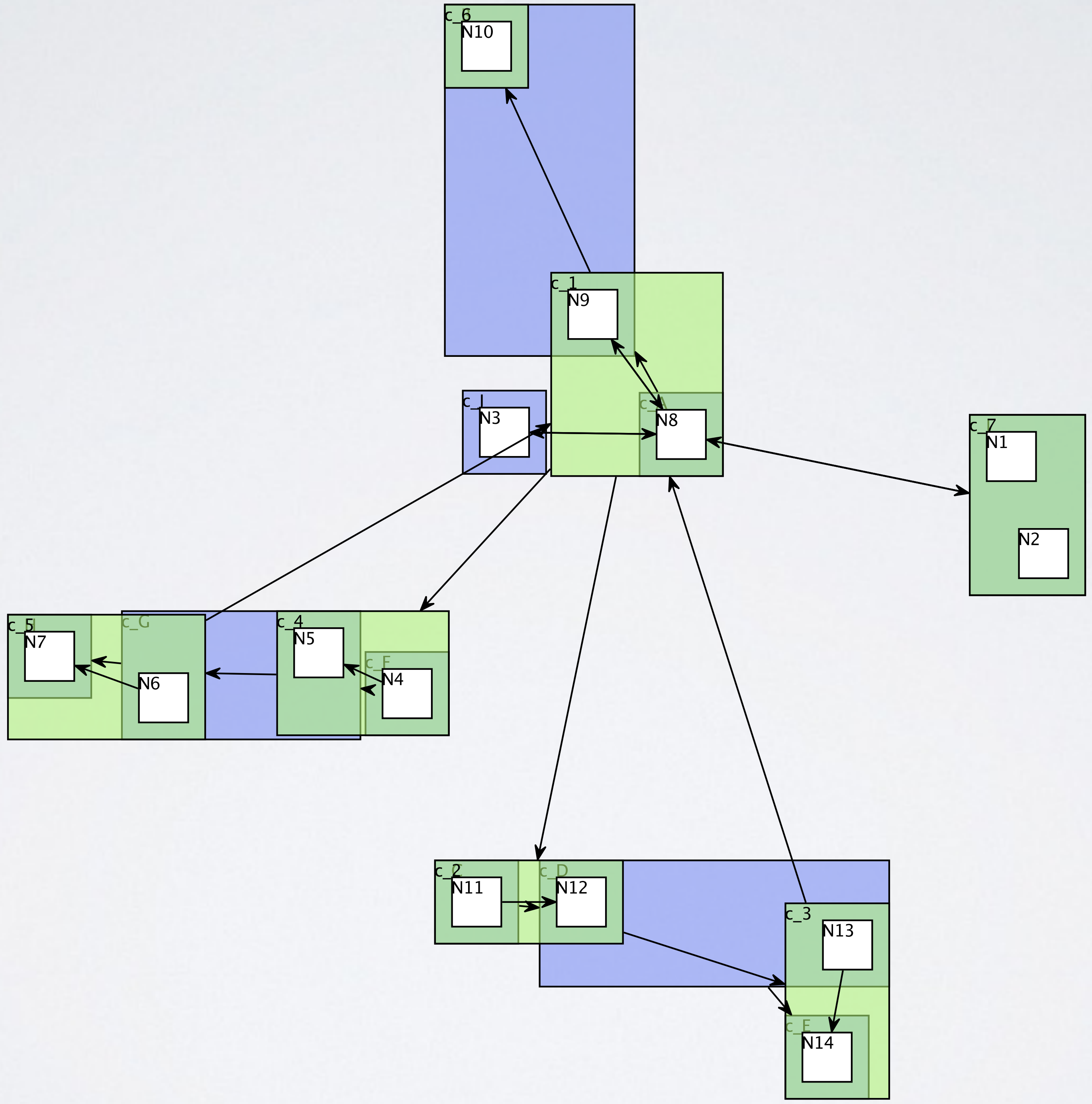
Manual Sample Layout



FORCE-BASED APPROACH



FORCE-BASED APPROACH



Next Steps

1. Drawability
   How can we find out whether we can actually draw a given Higraph?

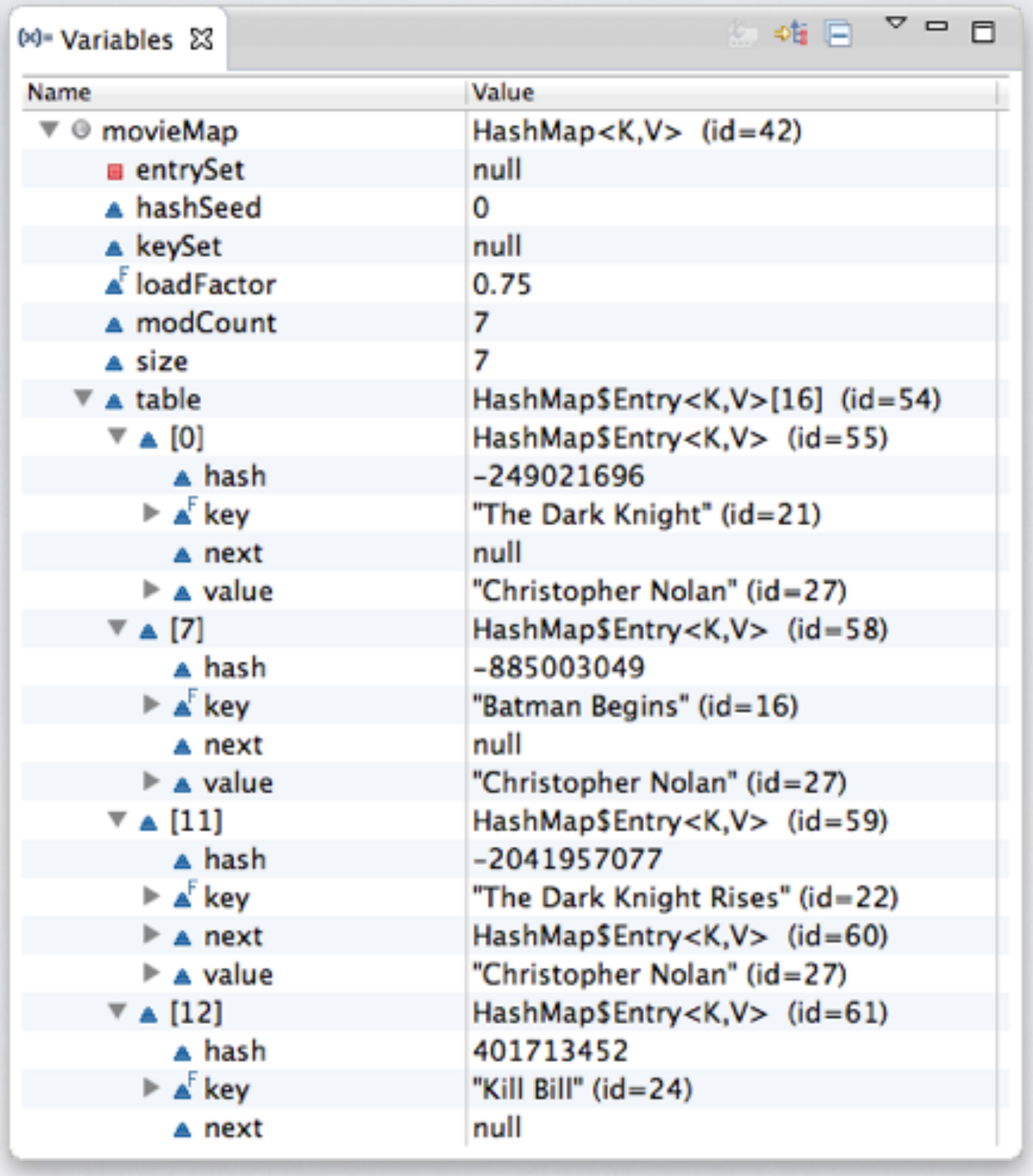
2. Algorithms
   How, then, can we draw Higraphs, or at least some Higraphs?

Koala!

GREETINGS FROM MELBOURNE

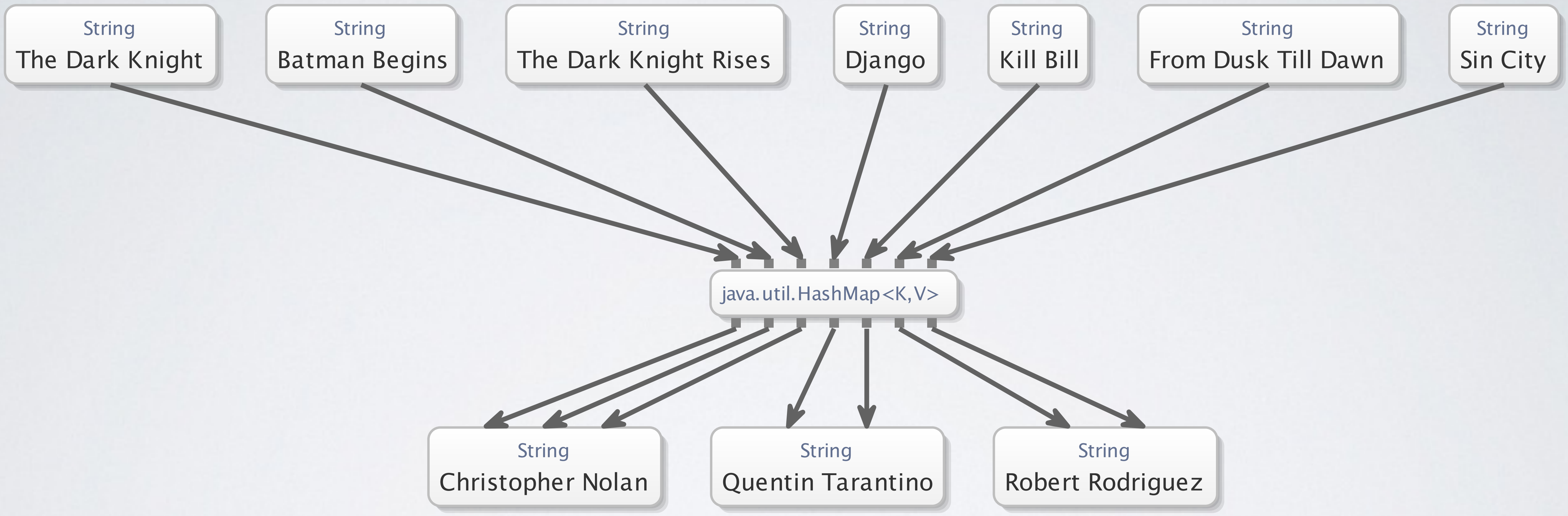
…and Stuff!

Our Friend, the Tree (View)



Name	Value
movieMap	HashMap<K,V> (id=42)
entrySet	null
hashSeed	0
keySet	null
loadFactor	0.75
modCount	7
size	7
table	HashMap$Entry<K,V>[16] (id=54)
[0]	HashMap$Entry<K,V> (id=55)
hash	-249021696
key	"The Dark Knight" (id=21)
next	null
value	"Christopher Nolan" (id=27)
[7]	HashMap$Entry<K,V> (id=58)
hash	-885003049
key	"Batman Begins" (id=16)
next	null
value	"Christopher Nolan" (id=27)
[11]	HashMap$Entry<K,V> (id=59)
hash	-2041957077
key	"The Dark Knight Rises" (id=22)
next	HashMap$Entry<K,V> (id=60)
value	"Christopher Nolan" (id=27)
[12]	HashMap$Entry<K,V> (id=61)
hash	401713452
key	"Kill Bill" (id=24)
next	null

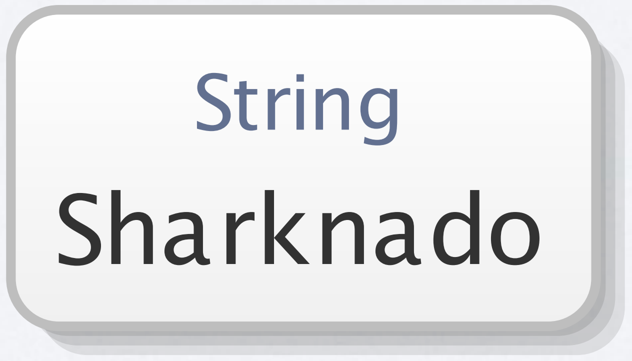
DEBUGGING WITH DEBUKVIZ



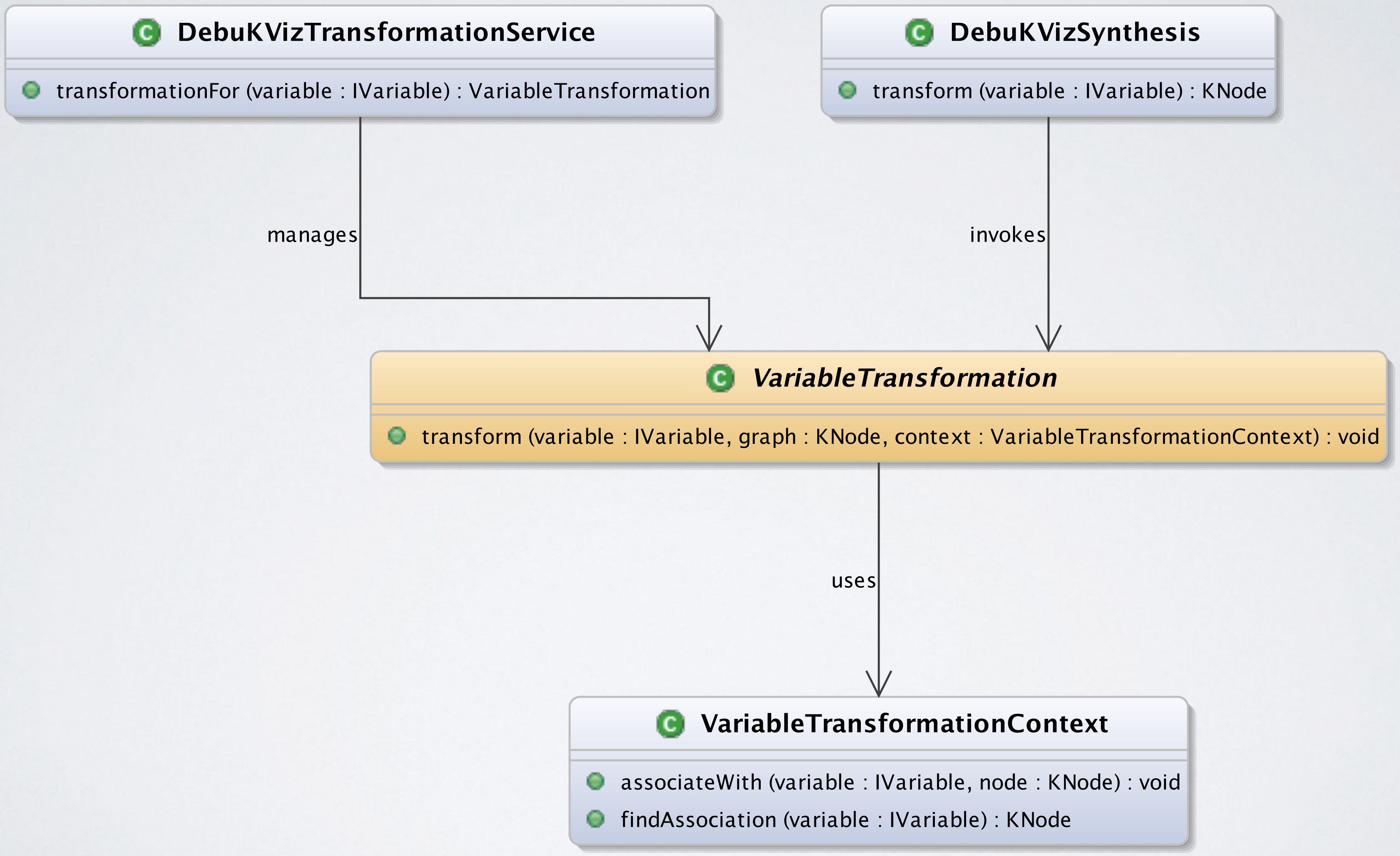
The String Transformation

```
class StringTransformation extends VariableTransformation {
    override transform(IVariable variable,
                       KNode graph,
                       VariableTransformationContext context) {

        NodeBuilder.forVariable(variable, graph, context)
            .type("String")
            .value(variable.stringValue)
            .build();
    }
}
```



DebuKViz Architecture


DebuKVizTransformationService
transformationFor (variable : IVariable) : VariableTransformation
DebuKVizSynthesis
transform (variable : IVariable) : KNode
manages
invokes
VariableTransformation
transform (variable : IVariable, graph : KNode, context : VariableTransformationContext) : void
uses
VariableTransformationContext
associateWith (variable : IVariable, node : KNode) : void
findAssociation (variable : IVariable) : KNode

Introducing OpenKieler

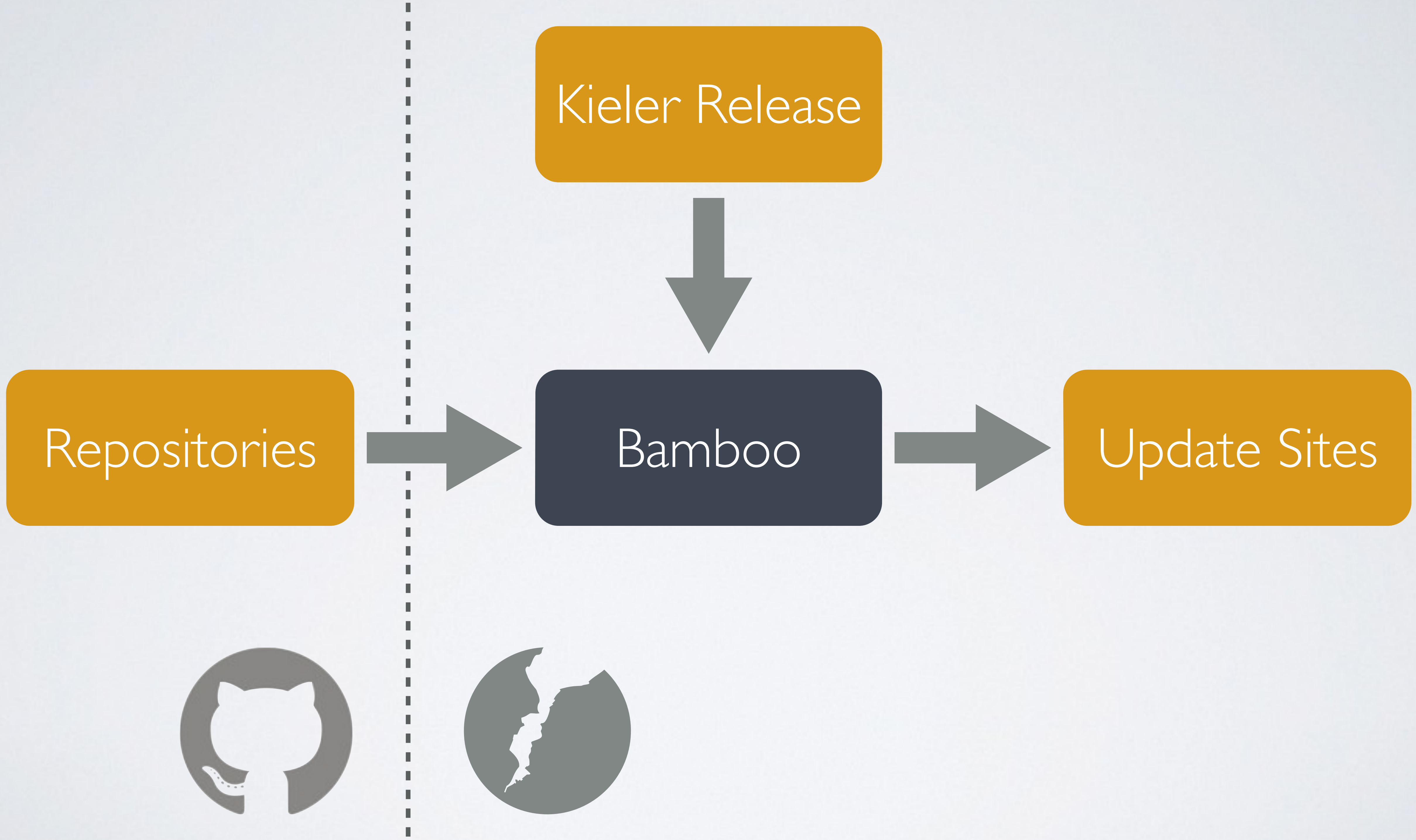
DebuKViz
Exploration of Java object trees when debugging

KlassViz
Ad-hoc visualization of Java class hierarchies

What is OpenKieler?

- Useful applications,
- with no research focus,
- built on Kieler,
- released as Open Source.

OpenKieler Infrastructure



Even More OpenKieler!

Heiko Wissmann

DebuKViz
Exploration of Java object trees during debugging

Enno Schwanke

KlassViz
Ad-hoc visualization of Java class hierarchies

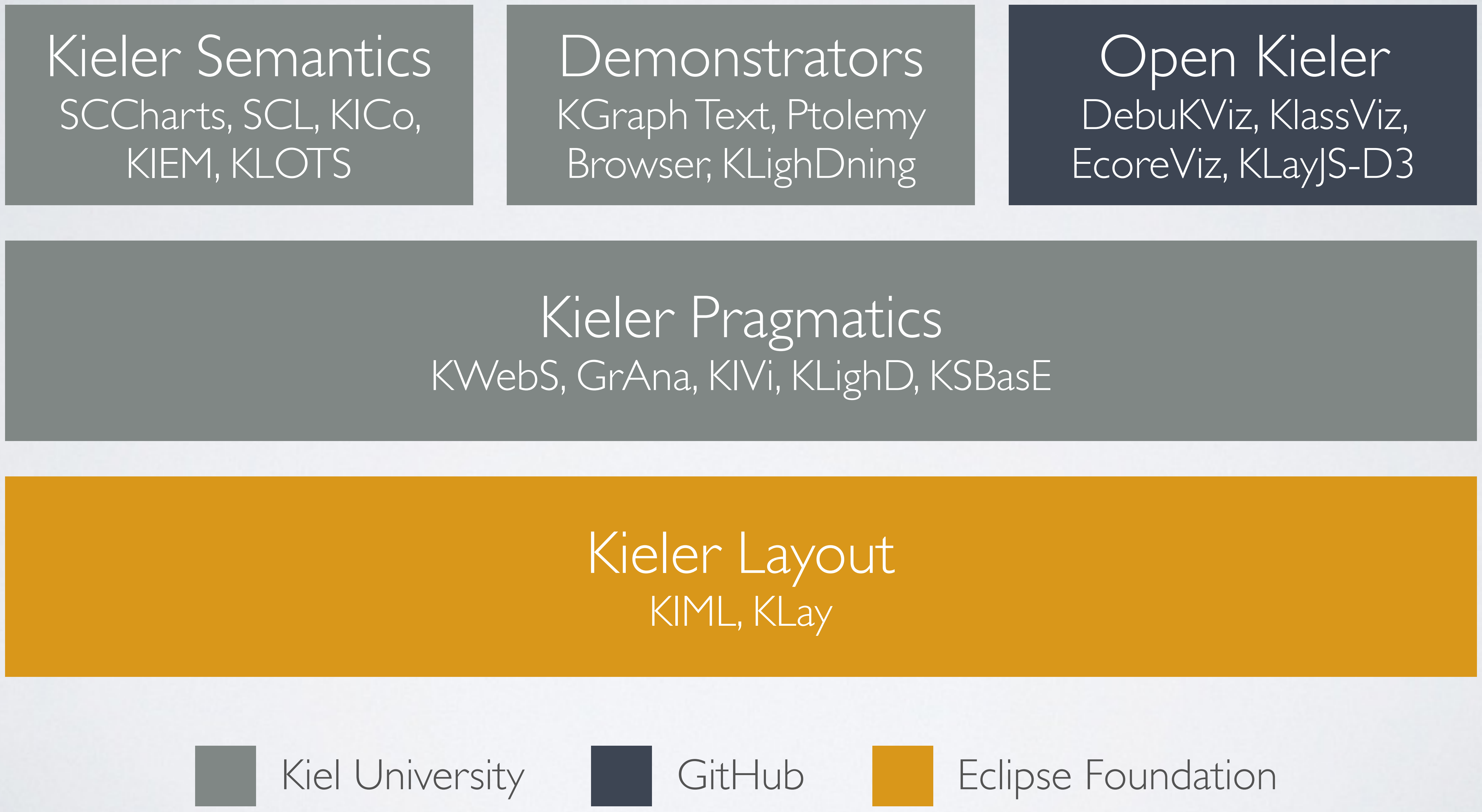
EcoreViz
Ad-hoc visualization of Ecore models

Schneidi

KLayJS-D3
Binding between KLayJS and the D3 graph library

Rüggi

RESTRUCTURING KIELER


Kieler Semantics
SCCharts, SCL, KICo, KIEM, KLOTS
Demonstrators
KGraph Text, Ptolemy Browser, KLighDning
Open Kieler
DebuKViz, KlassViz, EcoreViz, KLayJS-D3
Kieler Pragmatics
KWebS, GrAna, KIVi, KLighD, KSBasE
Kieler Layout
KIML, KLay
Kiel University
GitHub
Eclipse Foundation

Conclusion

- Higraphs
  - Force-based approach
  - Drawability
- Open Kieler
  - Projects
  - Structure

GREETINGS FROM KIEL

…hosted by Ulf Rüegg!