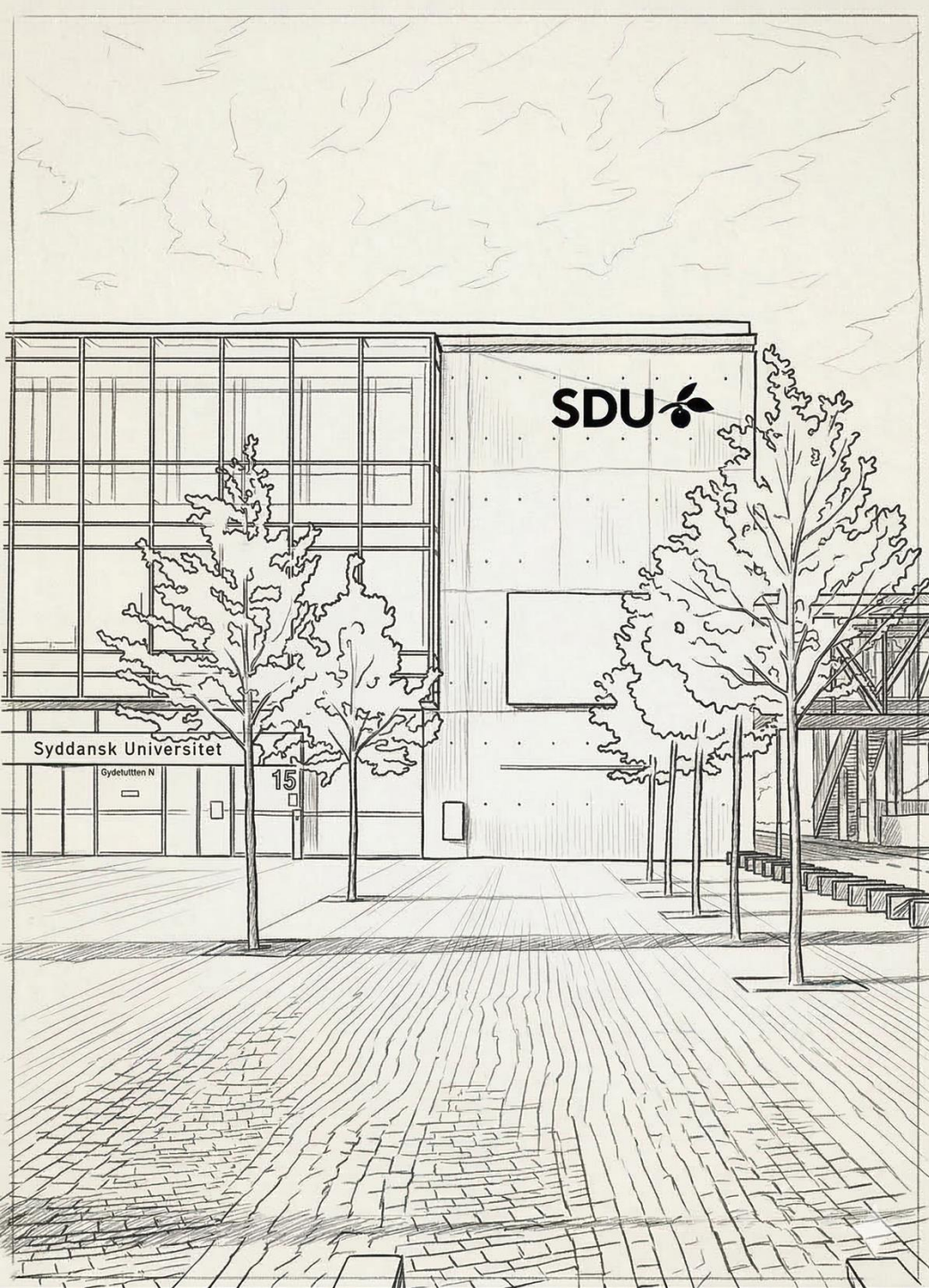




Towards Analysing N-Language Polyglot Programs

Breaking the two-languauge boundary in static analysis

Jyoti Prakash

Abhishek Tiwari

Mikkel Baun Kjærgaard

The Maersk McKinney Moller Institute

Programming in the Polyglot Era

Developers

Combine languages to harness specific strengths (C/C++ for hardware, Java or Kotlin for robustness, Python for AI/ML pipelines)

Result

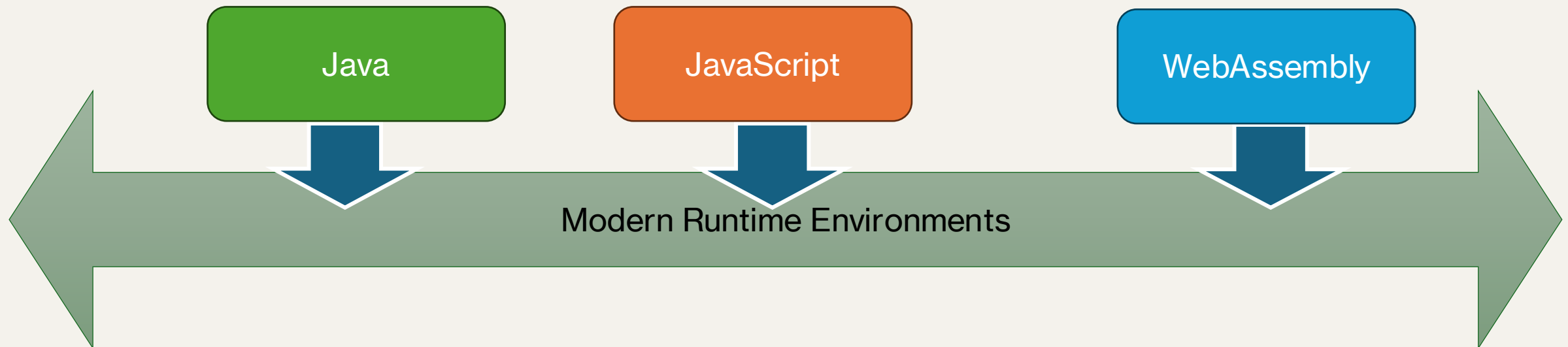
Complex programs combine diverse runtime environments

Runtimes

Make this integration feasible and widespread (GraalVM, Android NDK, WebView)

Challenge

Effective static and dynamic analysis of these programs



Two-Language Boundary

Current research successfully analyzes programs with two-languages interoperability

Analyses rely on isolated boundaries and one-to-one dataflow reconciliation. The problem appears solved

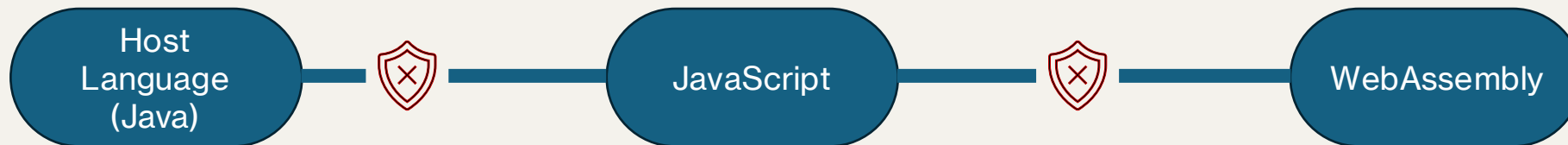


The Semantic Collapse

- Real-world systems now exceed the two-language boundary

**Extending analyses to three languages is not
a mere engineering inconvenience
it's a methodological gap**

- Dataflow traverse multiple runtime abstractions
- **New Semantic phone**

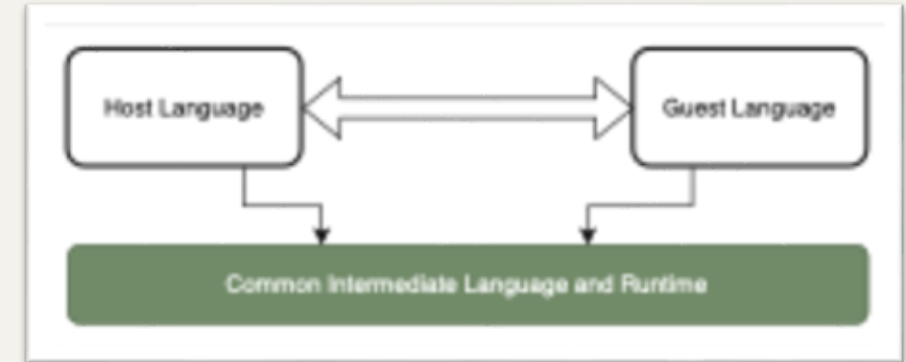


Three Archetypes of Polyglot Communication



Model 1

Foreign-Function Interfaces
across different runtimes



Model 2

Common Intermediate Representation
on a shared runtime



Model 3

Common Intermediate Representation
on a shared runtime

Challenge 1: Mutual Fixed Points

PROBLEMS

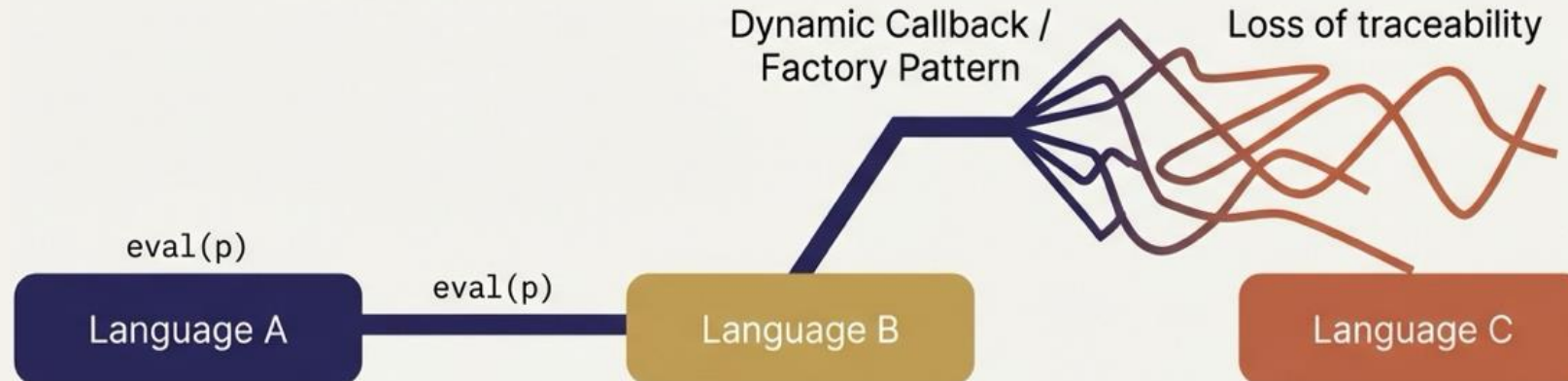
- Static Analysis on one language
Fixed-points over a **transfer function space**
- Two-languages
Fixed-points over a system of **a pair of transfer function spaces**
- Three-languages
Fixed-points over two systems of pairs transfer function spaces

PROPOSED DIRECTION

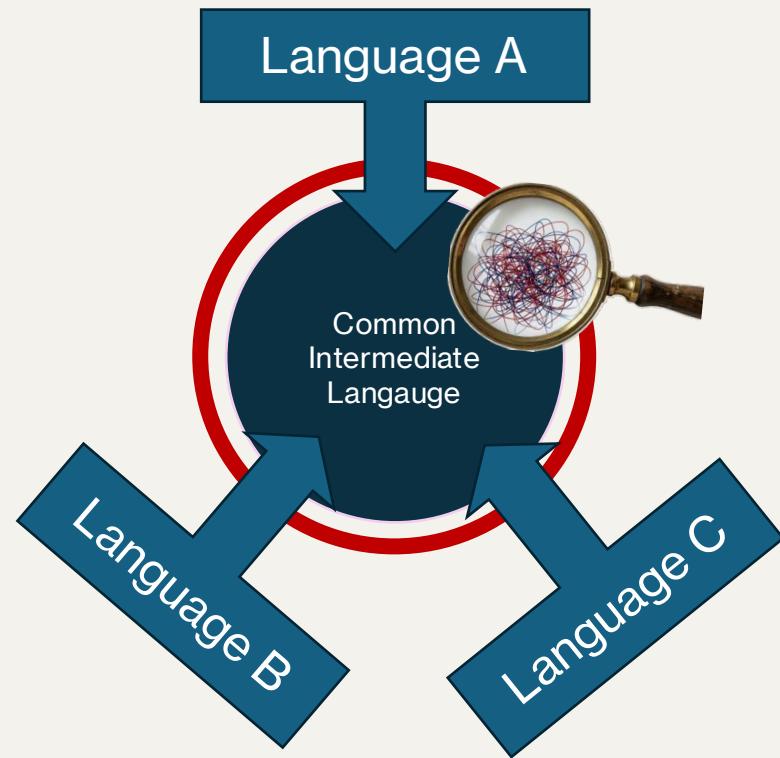
- Incremental propagation across differential transfer functions

Challenge 2: Multi-Hop Call Graphs

- Generating an inter-procedural call graph is not just adding a third language's edges
- **Problem:** Resolving cross-boundary dependencies requires processing APIs where the call target in Language A depends on dynamic choices made deep in the $B \rightarrow C$ chain
- **Takeway:** Dependencies are no longer point-to-points; they are multi-hop phenomena

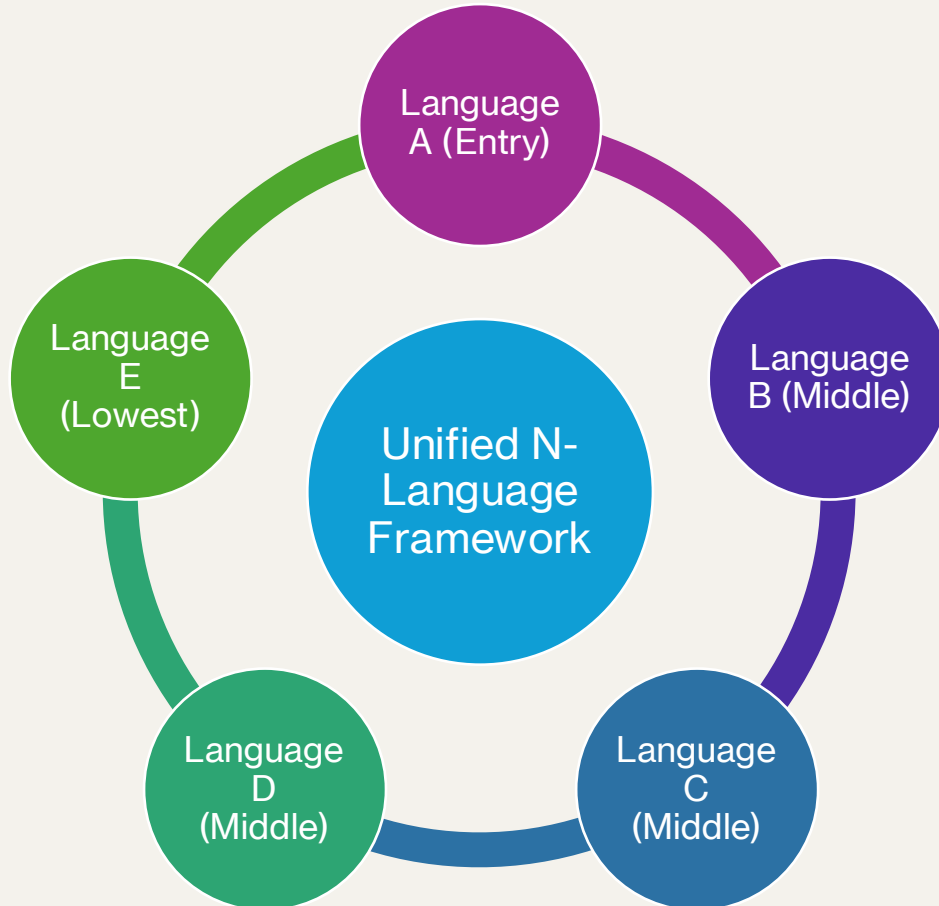


Challenge 3: The Illusion of Common IR



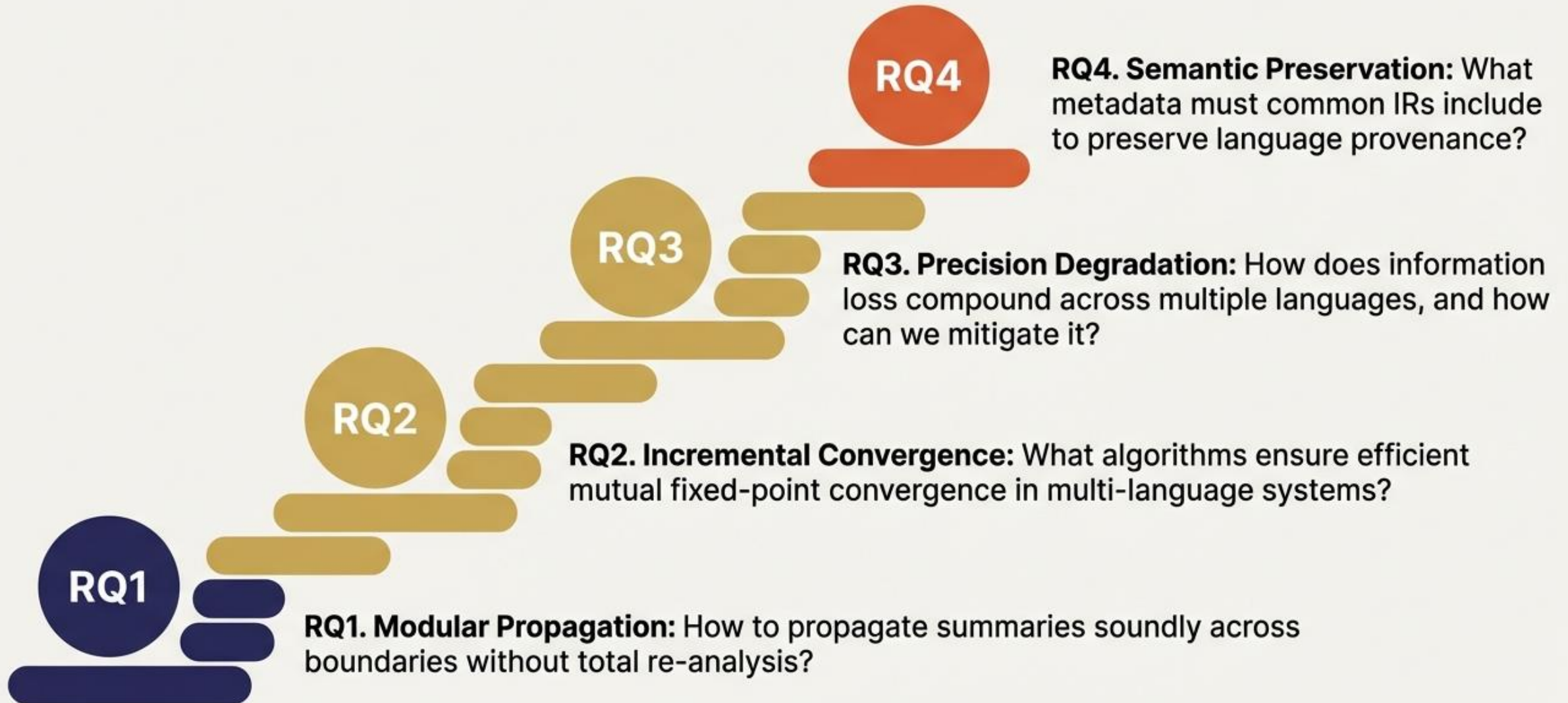
- Compiling multiple languages to shared IR eliminates memory model and calling convention complexities
- **Catch:** Constructing the call graph requires control flow analysis which is mutually recursive with data flow analysis
- Precise call-graphs requires multilingual dataflow analyses

Vision: N-Language Analysis



- Stop creating customised tools for every two-language combination
- **Goal:** Language agnostic framework capable of reasoning over entire multilingual eco-systems
- **Requirements**
 - Sound modular summaries
 - Incremental algorithms
 - Highly semantic intermediate representations

Roadmap for Future Research



Towards Analysing N-Language Polyglot Programs

