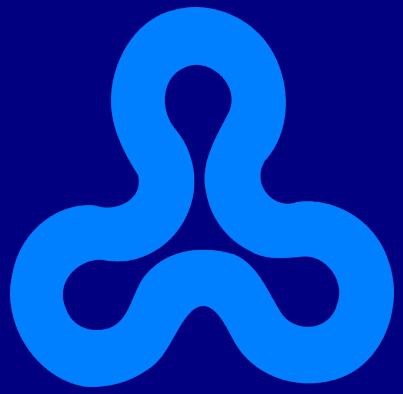
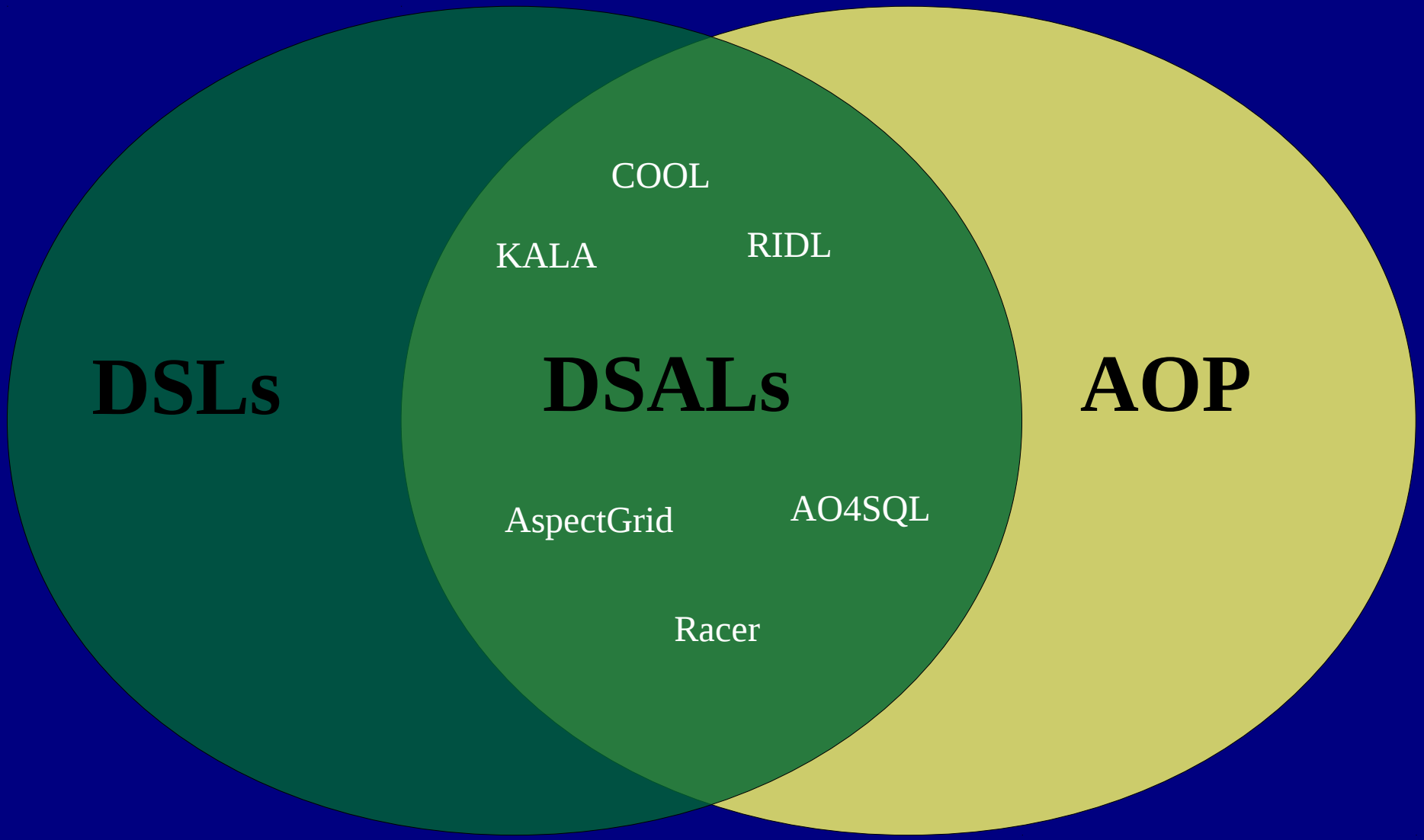
Demanding First-Class Equality for Domain Specific Aspect Languages

Arik Hadas
Dept. of Mathematics and Computer Science
The Open University of Israel



Joint Work With:
David H. Lorenz

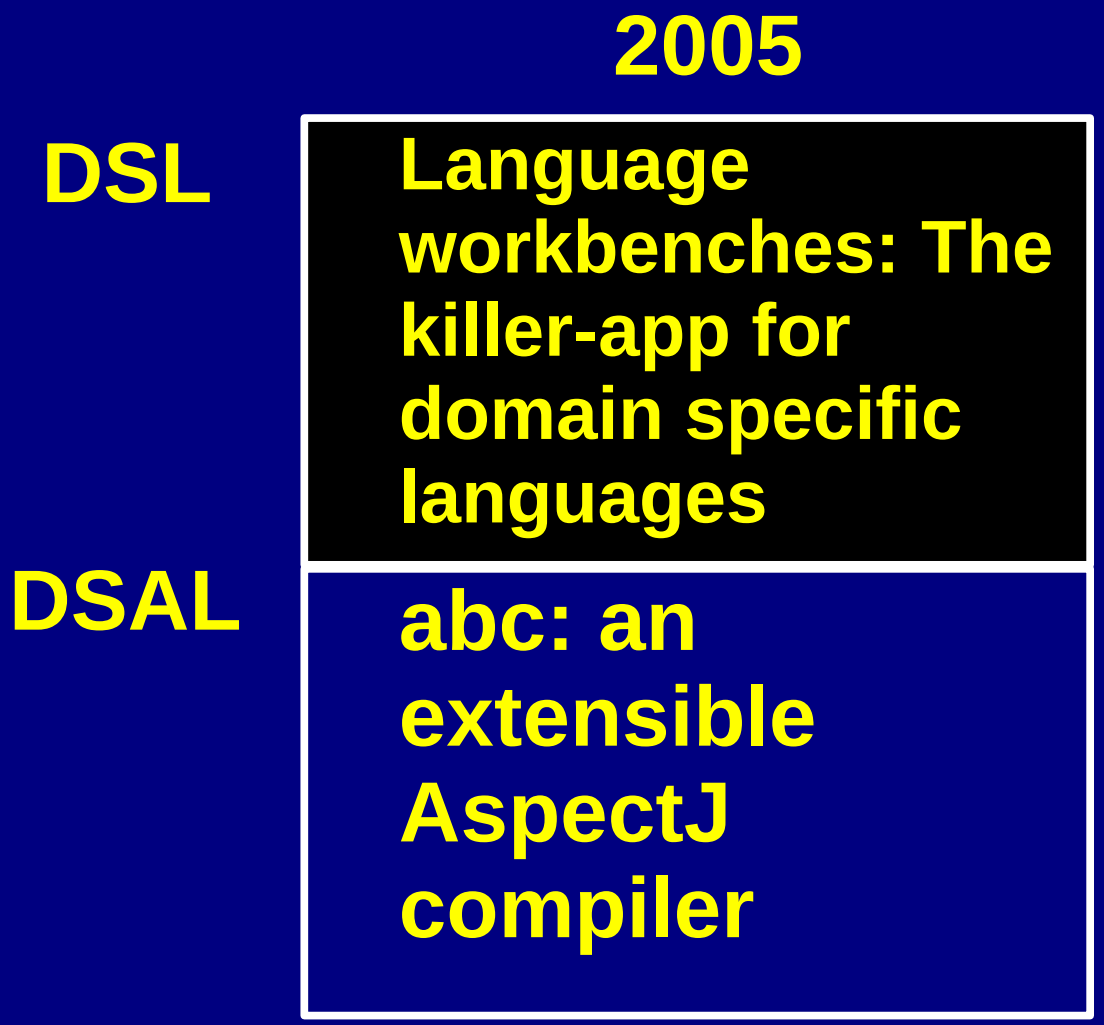
Domain Specific Aspect Languages



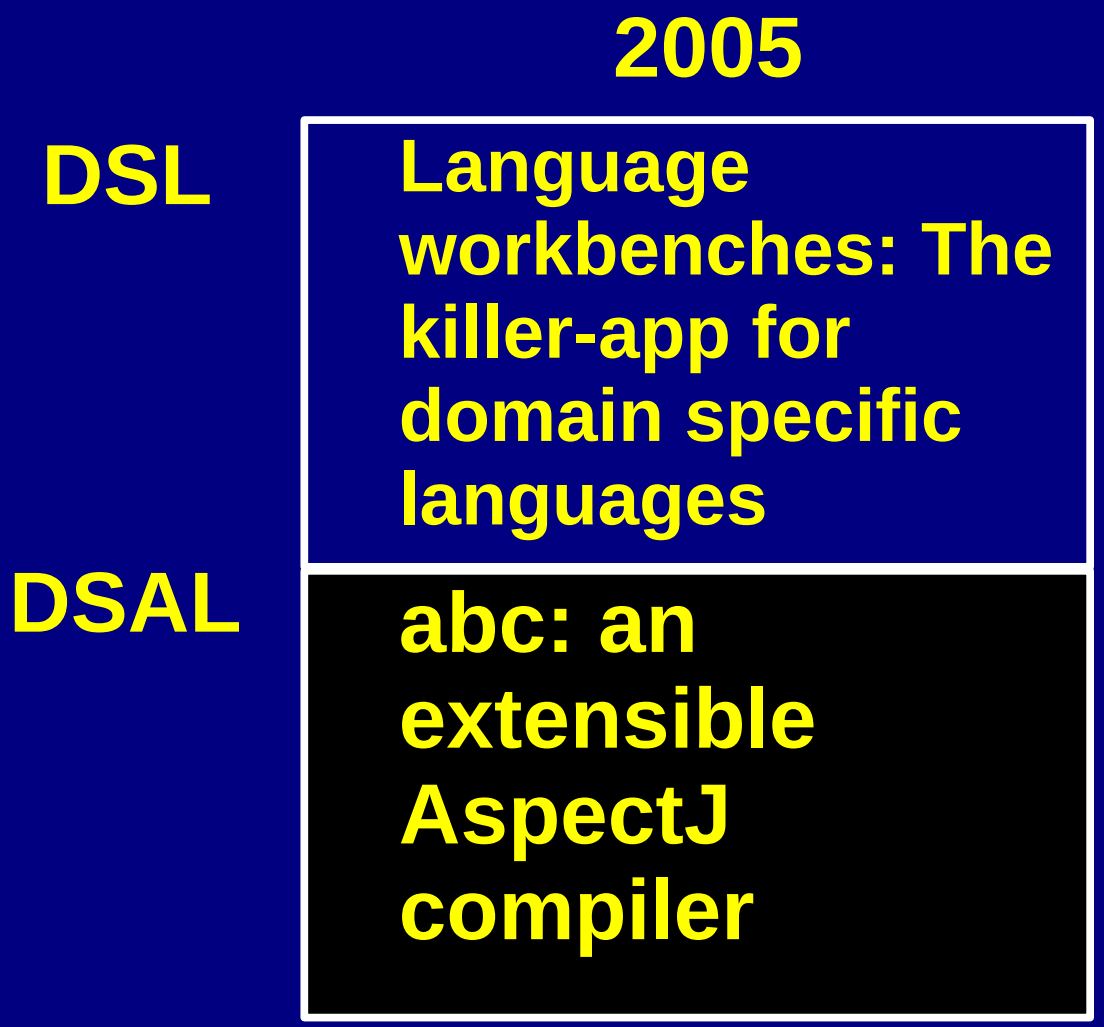
DSALs are Second-class

- **Second-class DSLs**
  - **Not as easy to develop and to use**
- **Second-class AOP languages**
  - **Incompatible with AOP tools**

Background


2005
DSL
Language workbenches: The killer-app for domain specific languages
DSAL
abc: an extensible AspectJ compiler

Background


2005
DSL
Language workbenches: The killer-app for domain specific languages
DSAL
abc: an extensible AspectJ compiler

Background

	2005	Since 2005
DSL	Language workbenches: The killer-app for domain specific languages	Language workbenches (LW) – xtext, MPS, Spoofax
DSAL	abc: an extensible AspectJ compiler	AOP composition frameworks (CF) – Reflex, XAspects, AWESOME

Background

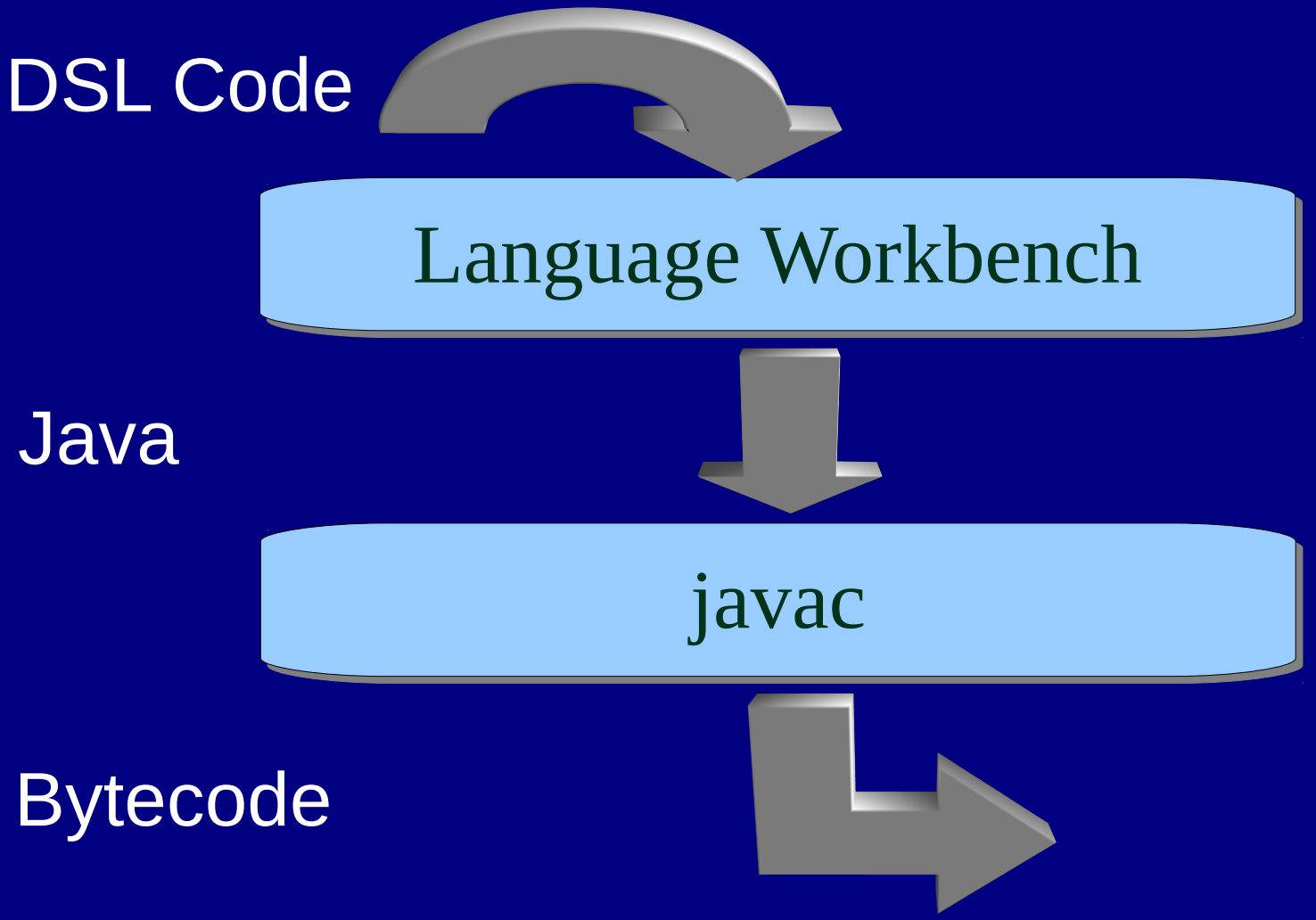
	2005	Since 2005
DSL	Language workbenches: The killer-app for domain specific languages	Language workbenches (LW) – xtext, MPS, Spoofax
DSAL	abc: an extensible AspectJ compiler	AOP composition frameworks (CF) – Reflex, XAspects, AWESOME

Background

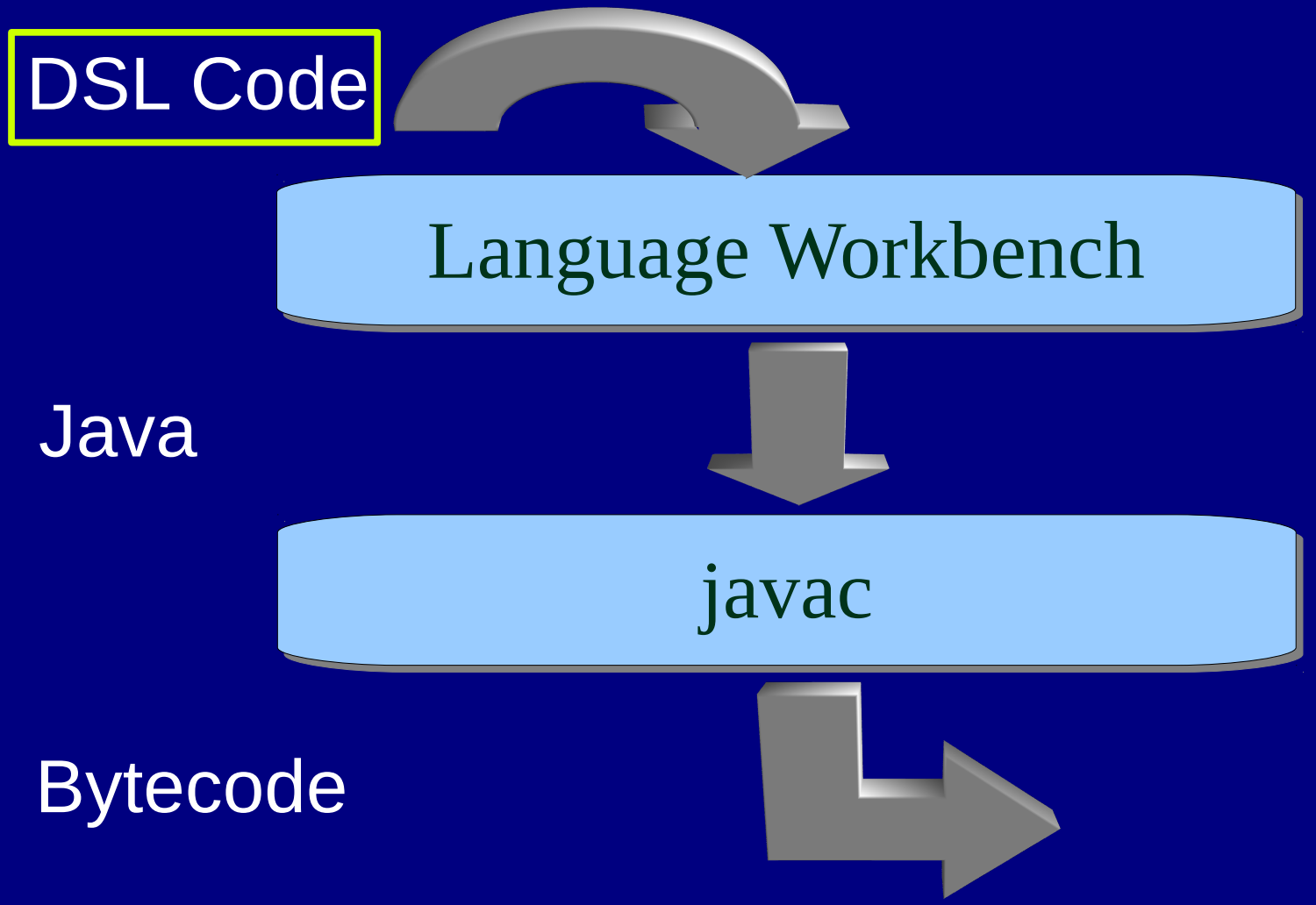
	2005	Since 2005
DSL	Language workbenches: The killer-app for domain specific languages	Language workbenches (LW) – xtext, MPS, Spoofax
DSAL	abc: an extensible AspectJ compiler	AOP composition frameworks (CF) – Reflex, XAspects, AWESOME

But no workbench solution for DSALs

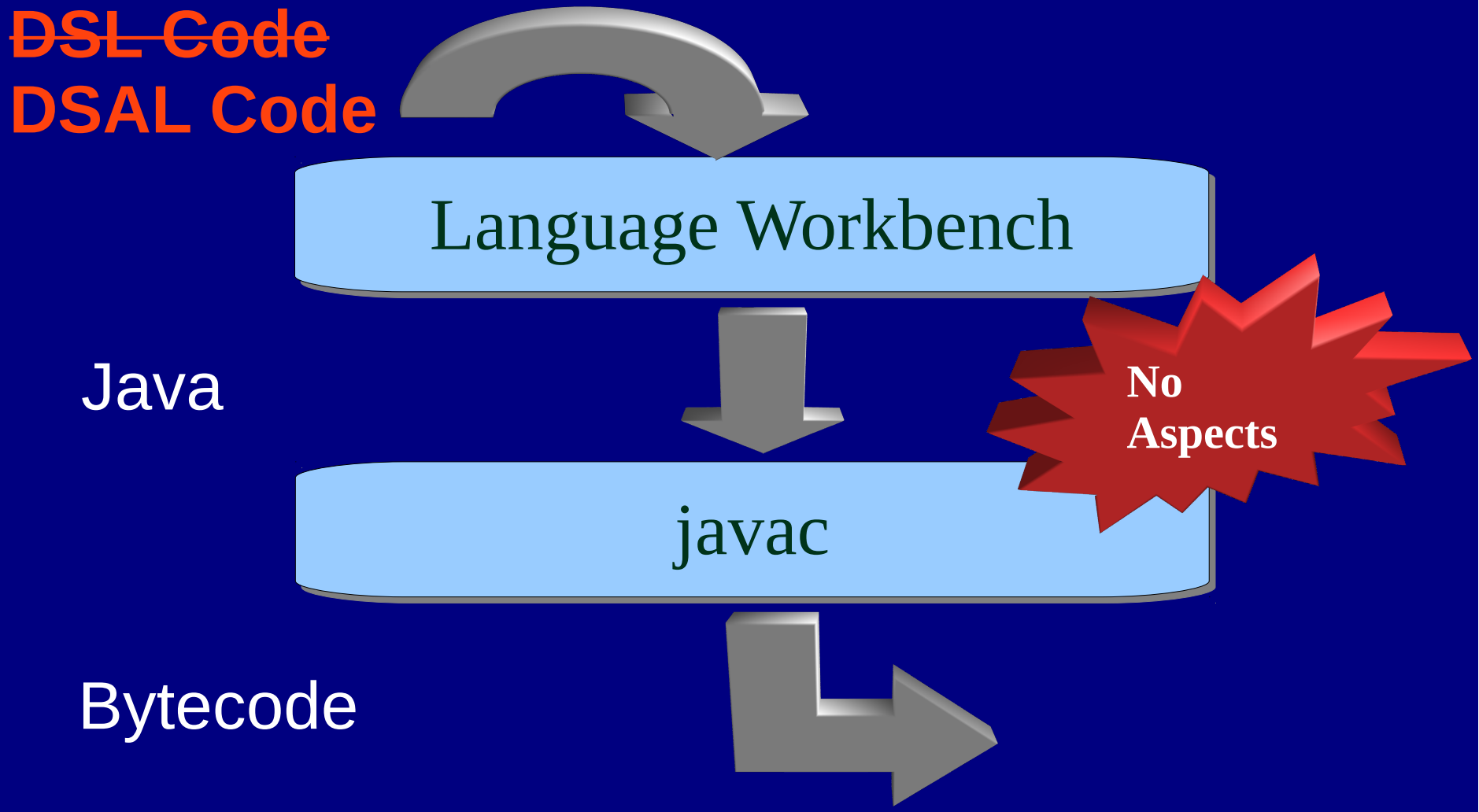
Language Workbench (LW)



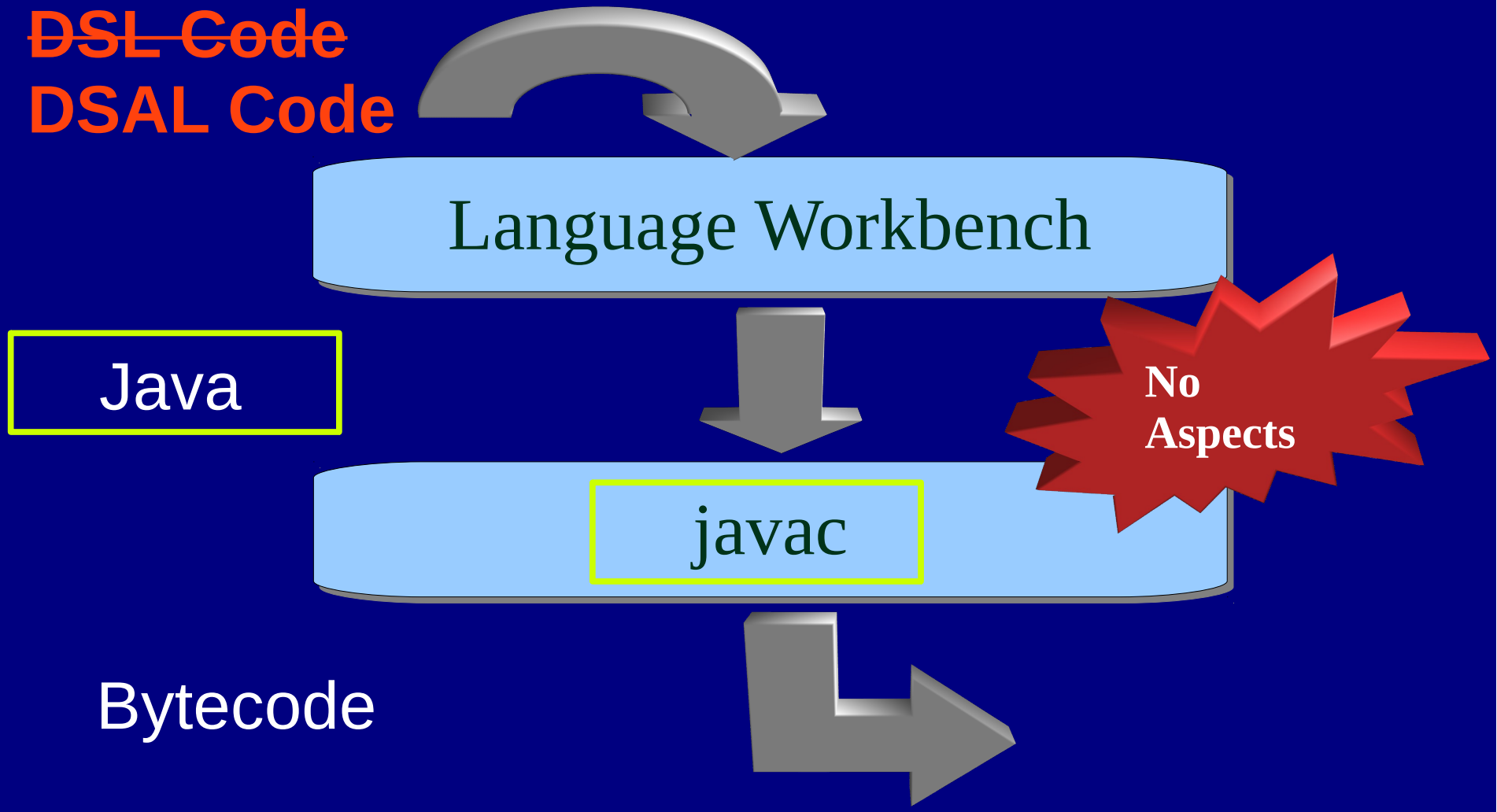
Language Workbench (LW)



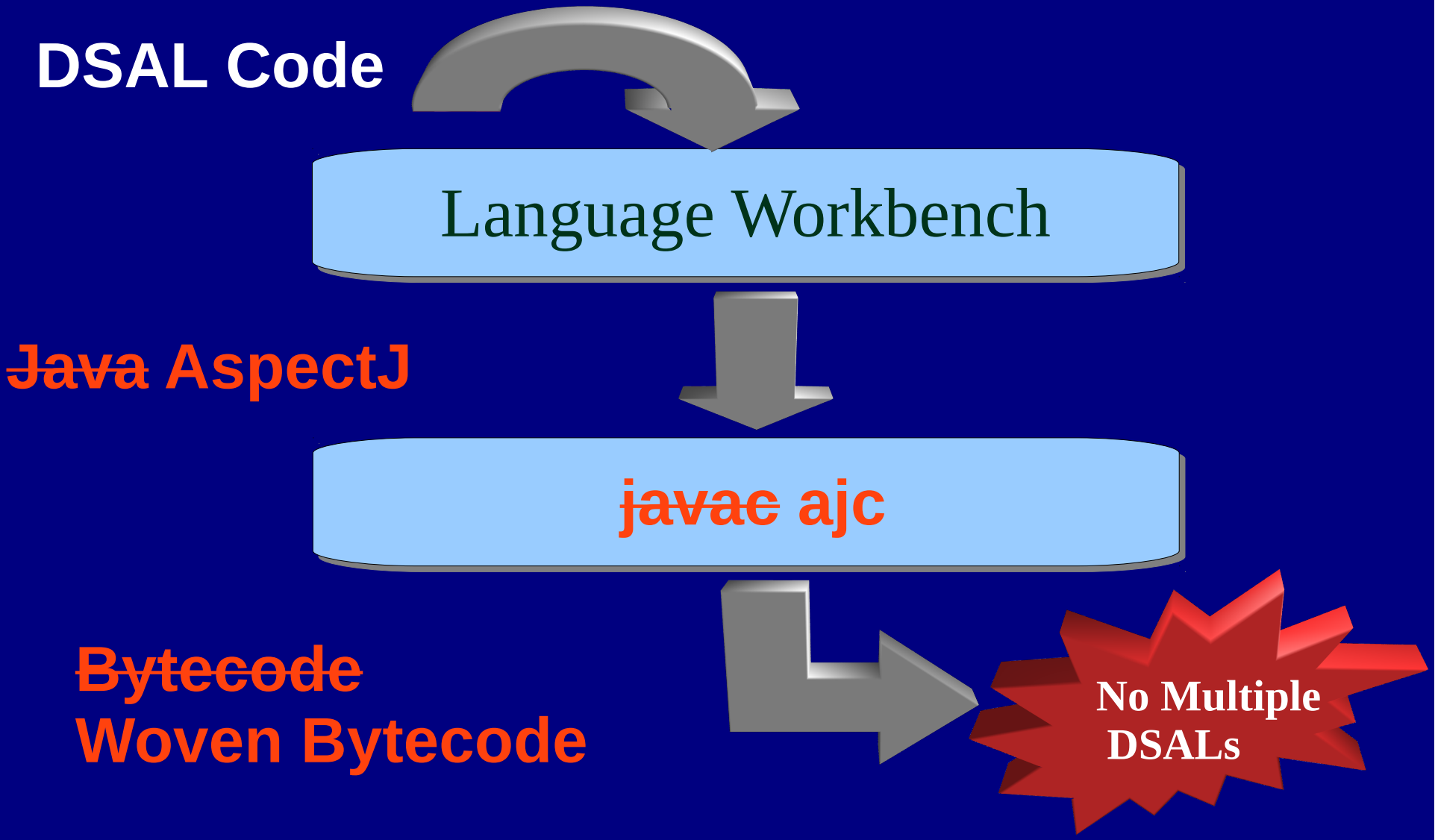
Using LW for DSAL



Using LW for DSAL

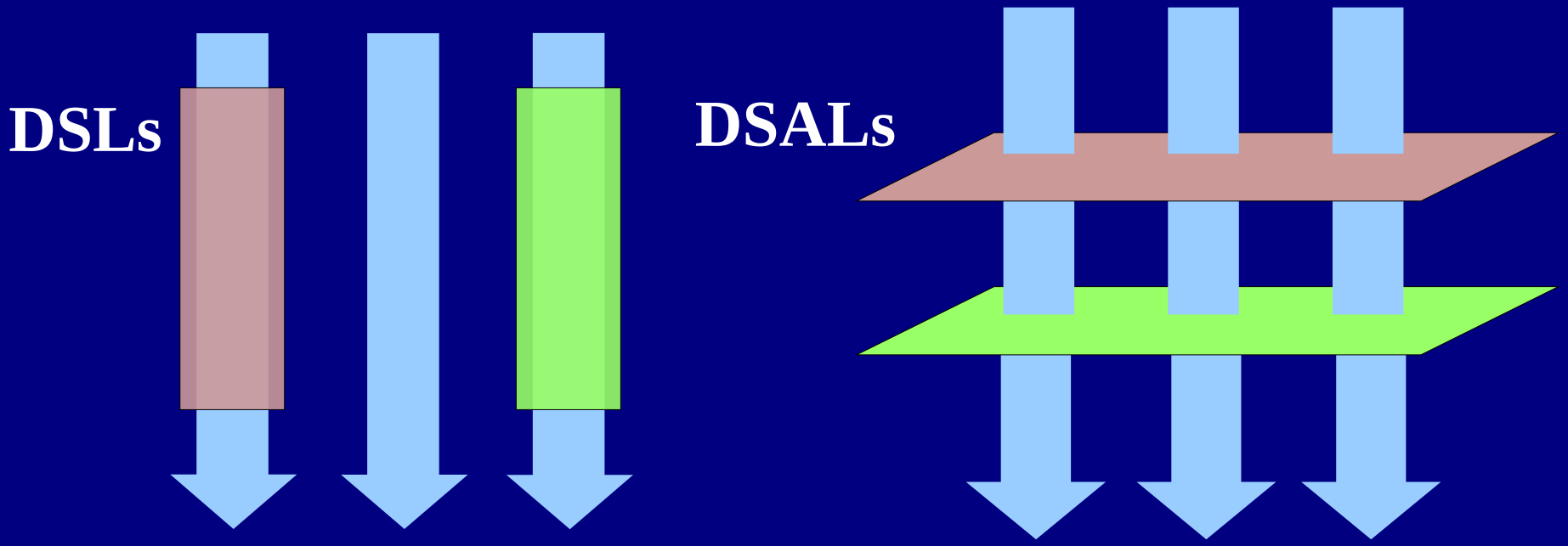


LW for AspectJ



DSAL Complexity

- **Unlike DSL, DSAL tackles crosscutting concerns**



Looking for a DSAL Workbench

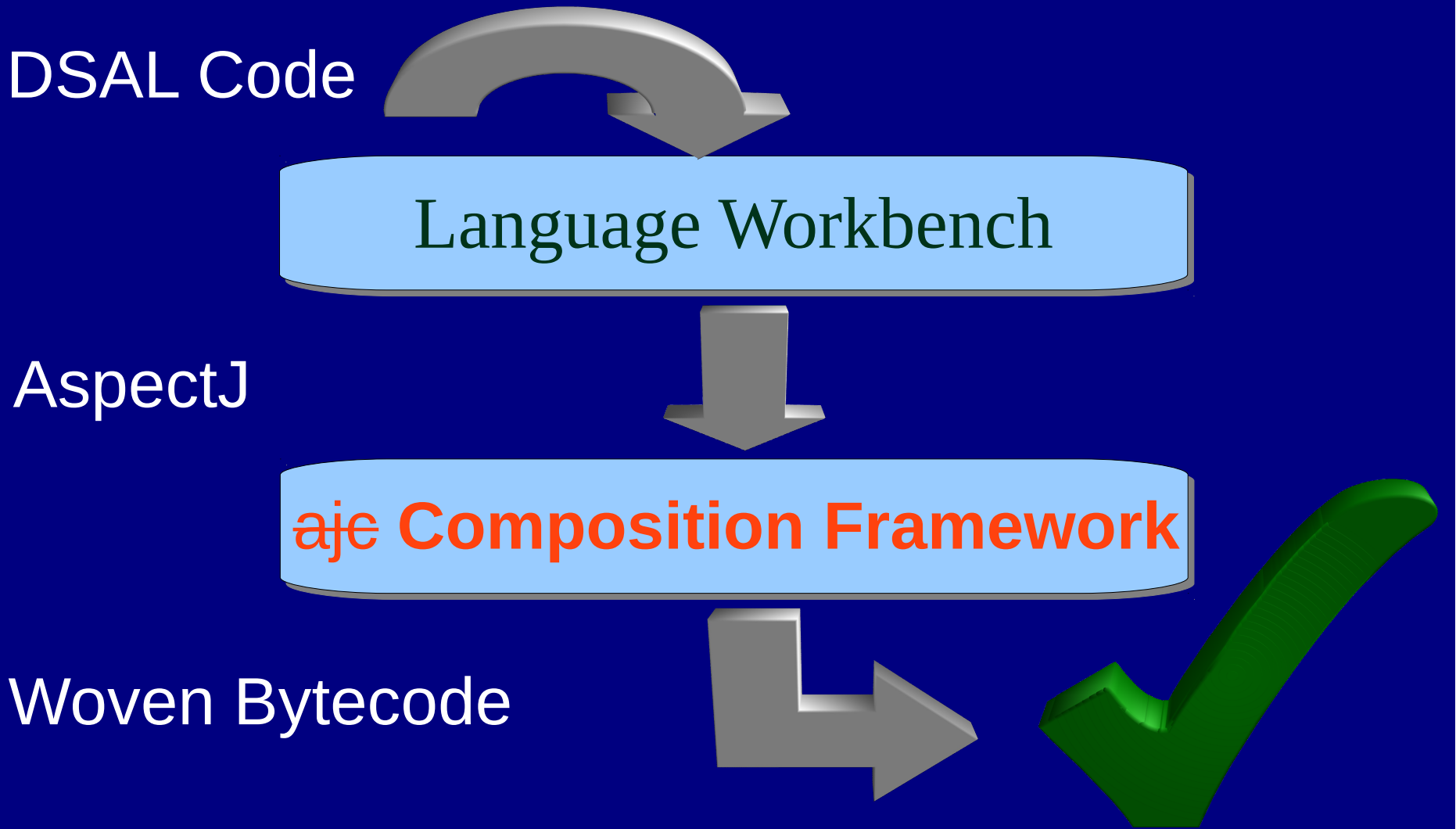
	Language Workbench	AOP Composition Framework
Tools for creation & usage of languages	☺	☹
Defining weaving semantics needed for DSALs	☹	☺

Looking for a DSAL Workbench

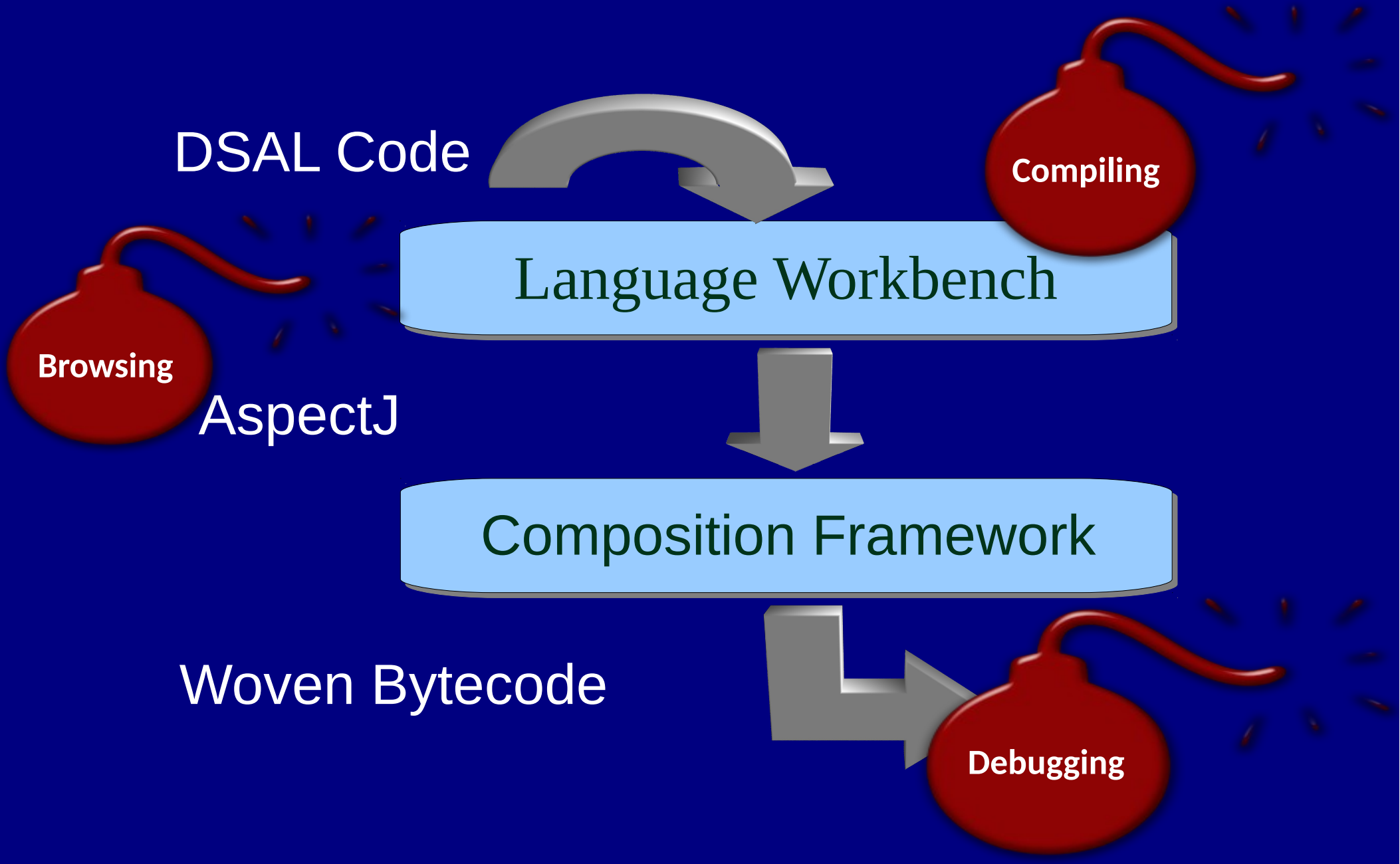
	Language Workbench	AOP Composition Framework
Tools for creation & usage of languages	☺	☹
Defining weaving semantics needed for DSALs	☹	☺

Will a naive combination of the two be a proper solution?

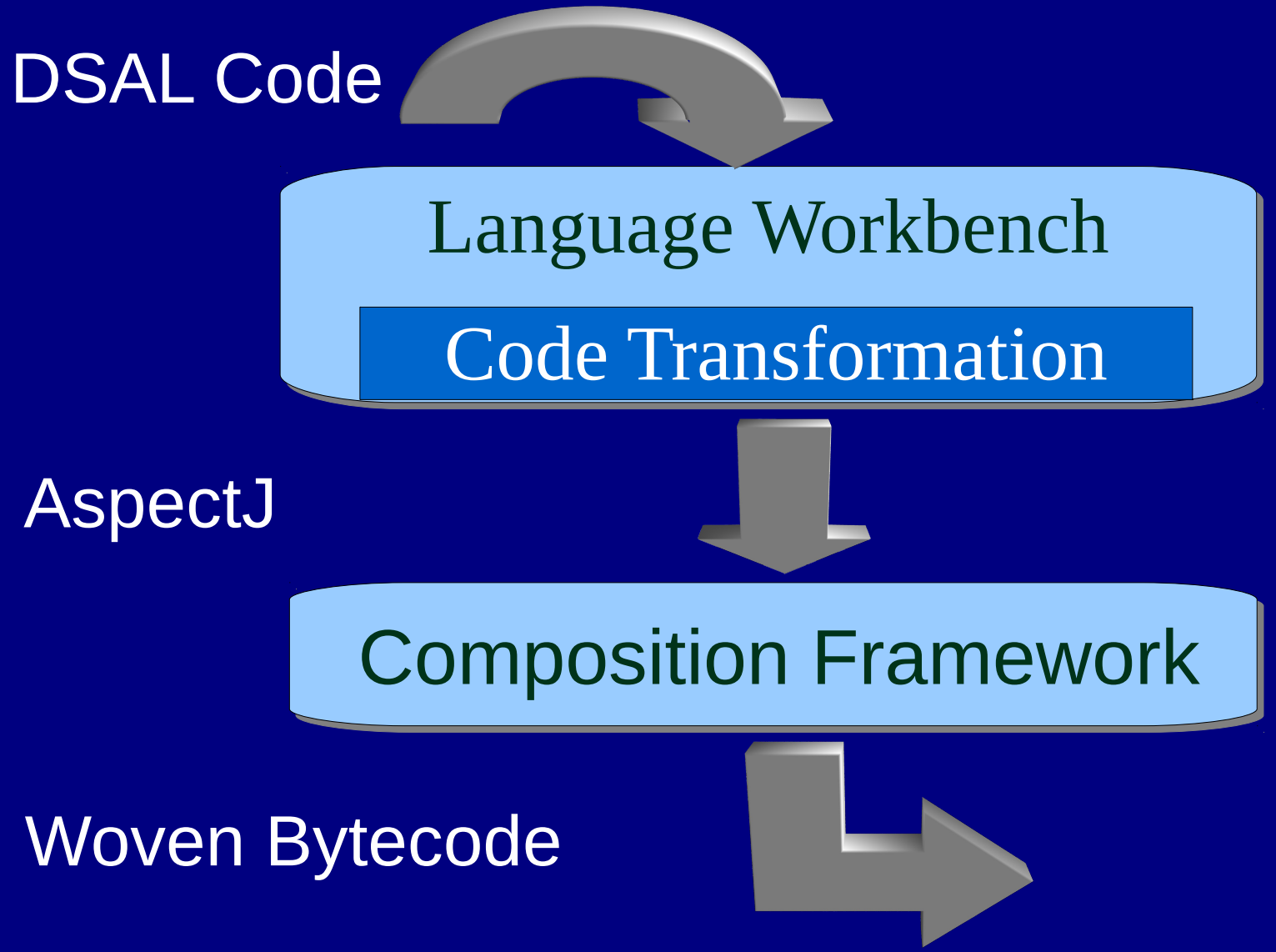
Naive Combination of LW and CF



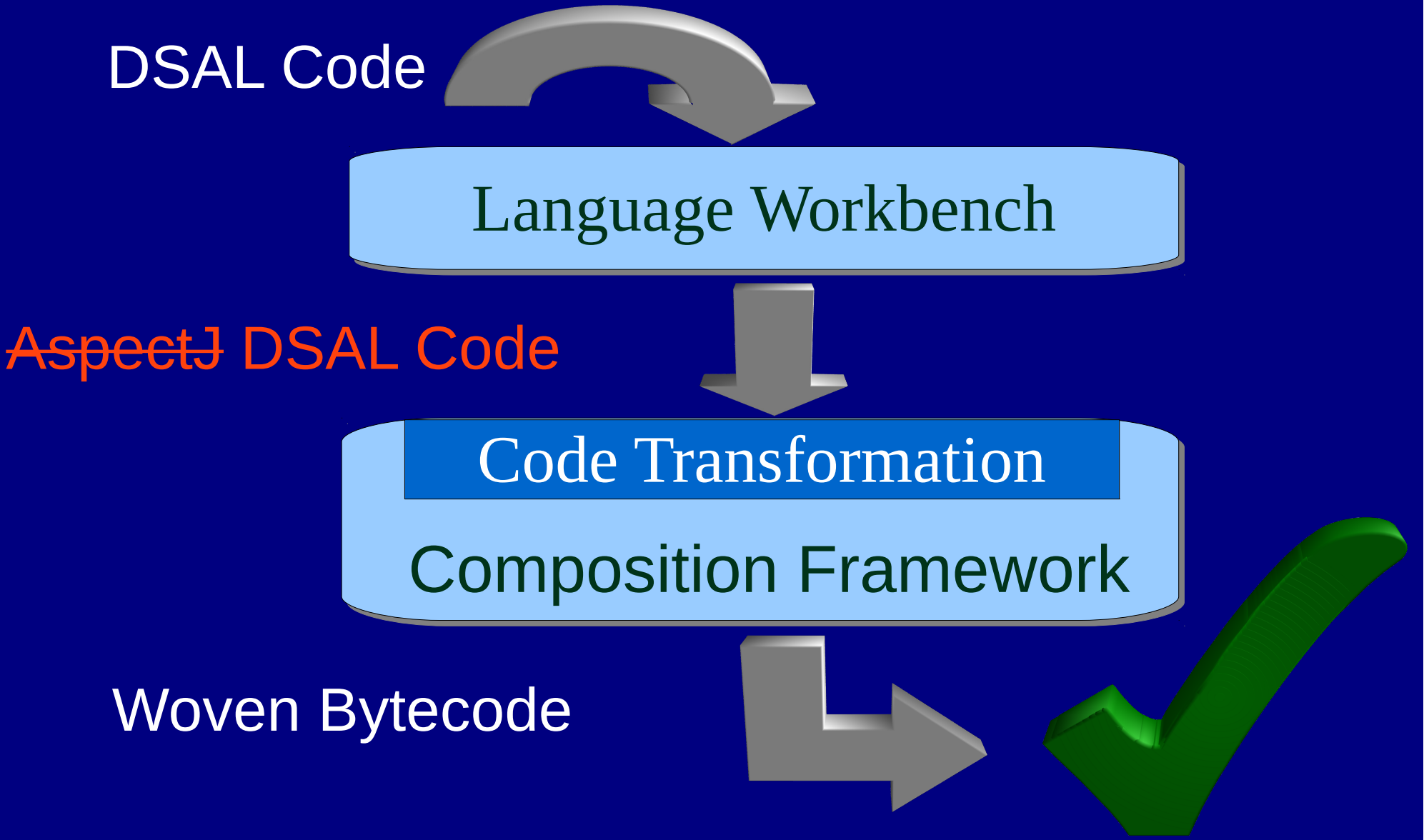
But Still Second-class AOP



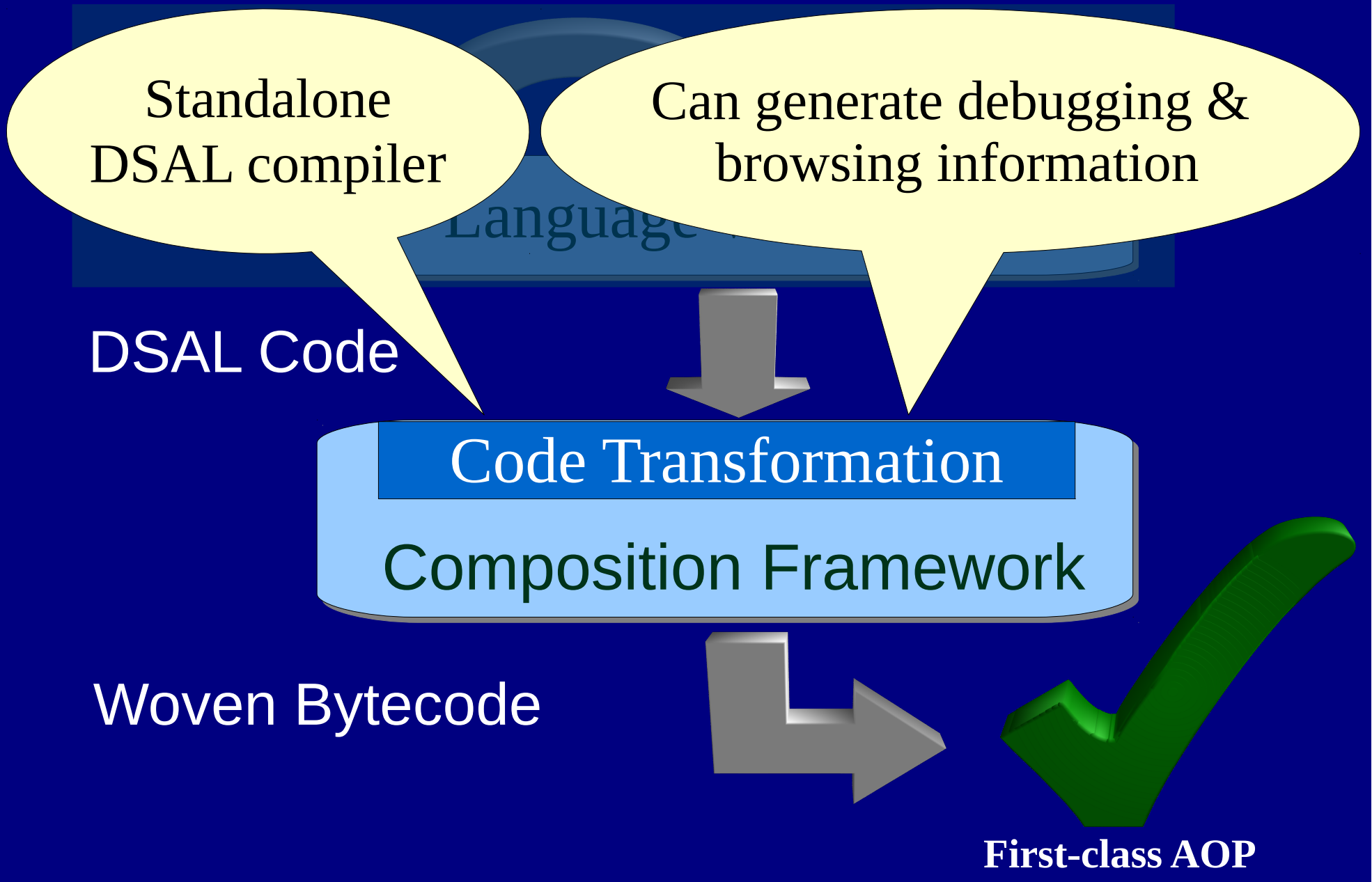
Traditional LW Architecture



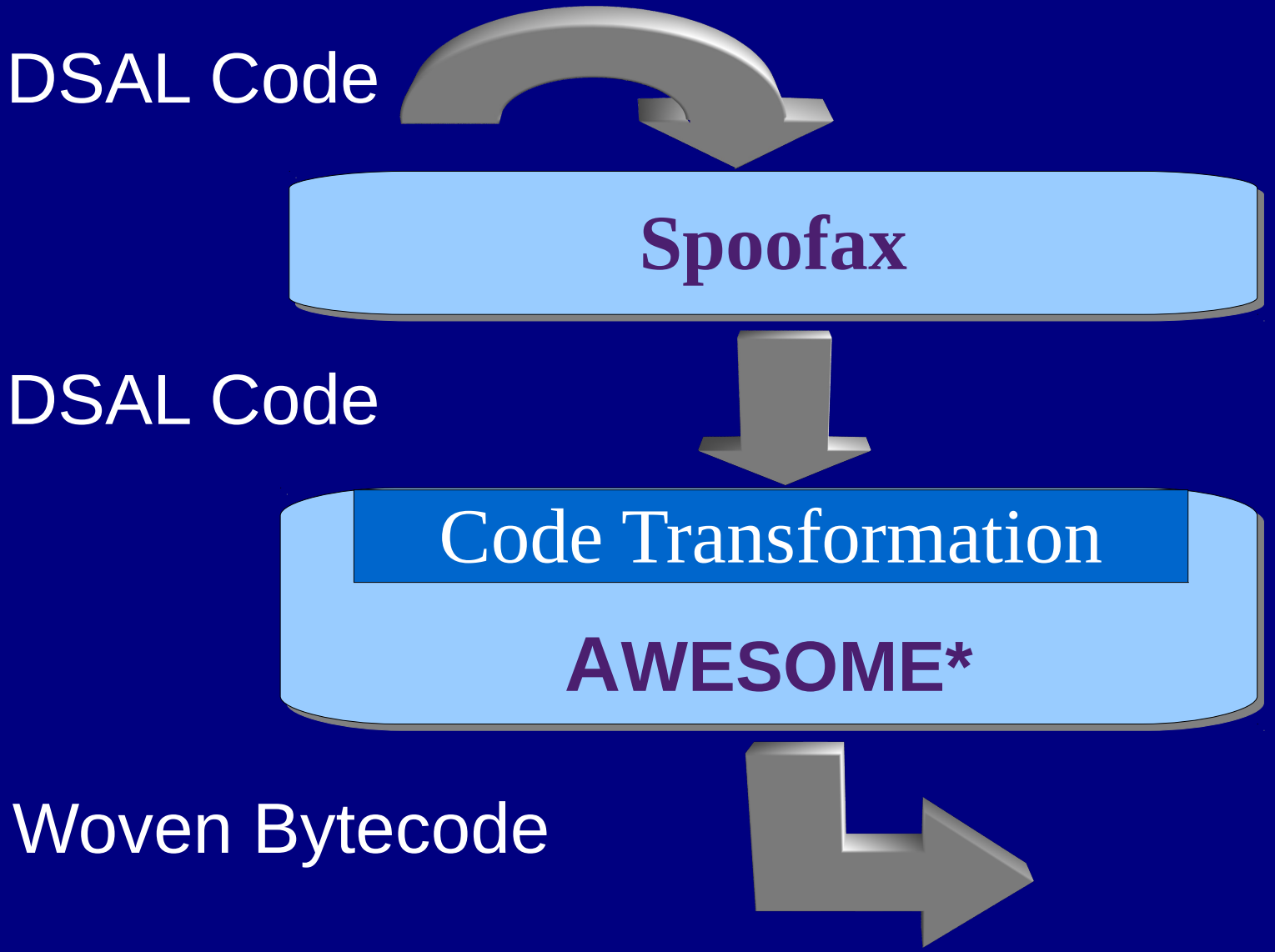
DSAL Workbench Architecture



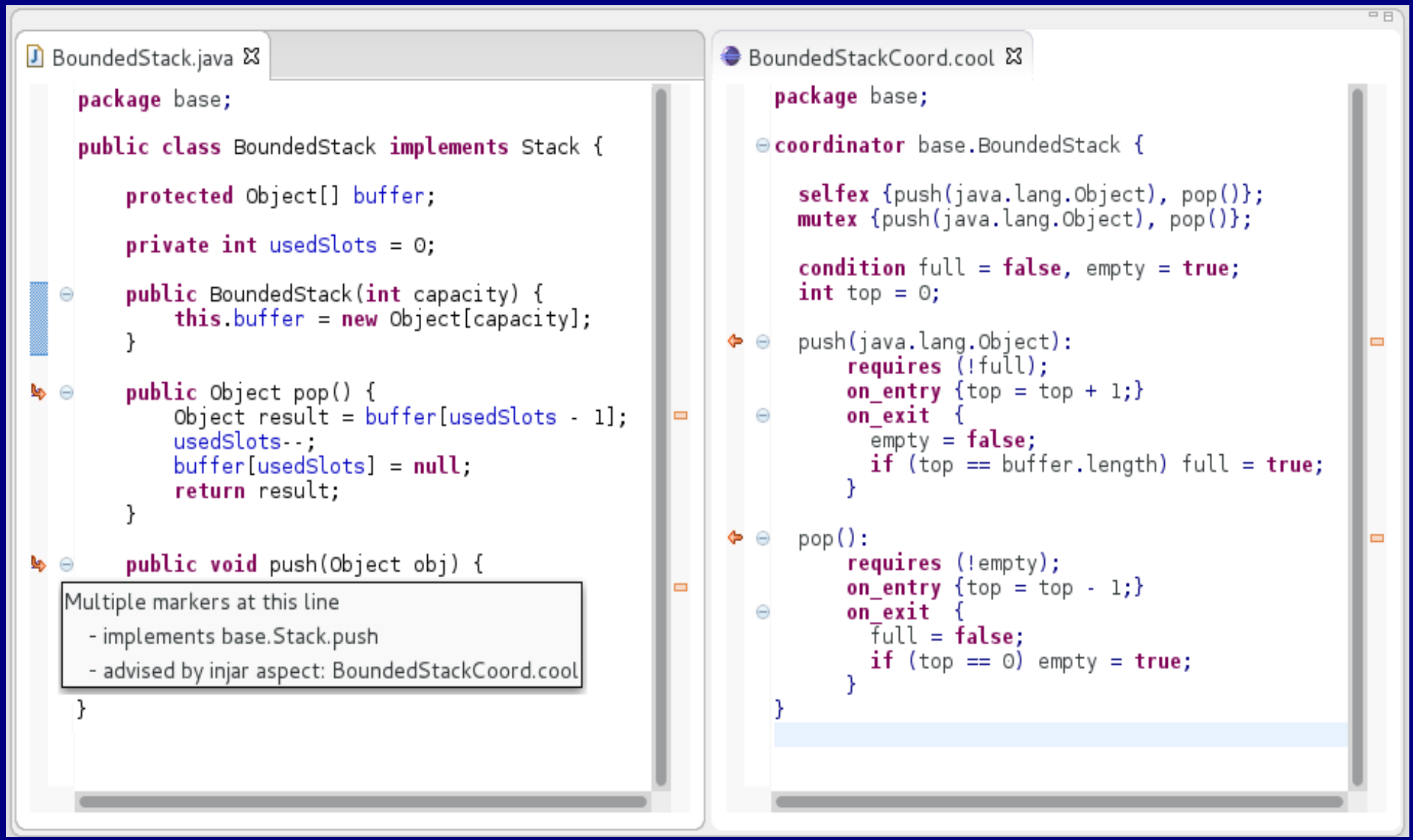
DSAL Workbench Architecture



Implementation



Example: First Class COOL



DSAL "Bill of Rights"

- *Freedom of Expression*
  - *Syntactic*
  - *Semantic*
- *Economic Freedom*
  - *Cost effective Implementation*
  - *Cost effective Usage*
- *Freedom of Assembly*
  - *DSL Interoperability*
- ***Equality with domain-specific languages and AOP languages***

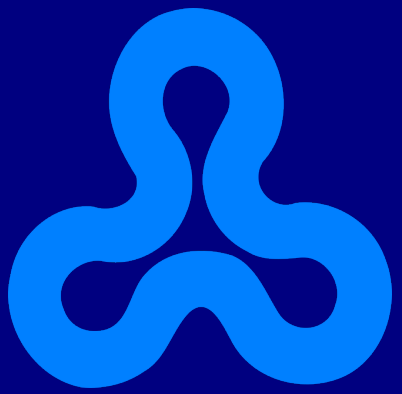
Related Work

- **Language Workbenches**
  - **[Fowler, 2005] Language workbenches: The killer-app for domain specific languages.**
  - **[Kats and Visser, 2005] The Spoofax language workbench: Rules for declarative specification of languages and IDEs.**
  - **[Lorenz and Rosenan, 2011] Cedalion: A language for language oriented programming.**
- **AOP Composition Frameworks**
  - **[Kojarski and Lorenz, 2005] Pluggable AOP: Designing aspect mechanisms for third-party composition.**
  - **[Lorenz and Kojarski, 2007] Understanding aspect interaction, co-advising and foreign advising.**
  - **[Kojarski and Lorenz, 2007] Awesome: An aspect co-weaving system for composing multiple aspect-oriented extensions.**

Conclusion

- A novel design for DSAL workbench that produces first-class DSAL
  - First-class DSL
  - First-class AOP language
- Prototype comprising Spoofax and AWESOME*
- Plug-in for COOL as a first-class DSAL

Thank You!



Arik Hadas and **David H. Lorenz**
Dept. of Mathematics and Computer Science
The Open University of Israel

arik.hadas@openu.ac.il

https://github.com/OpenUniversity