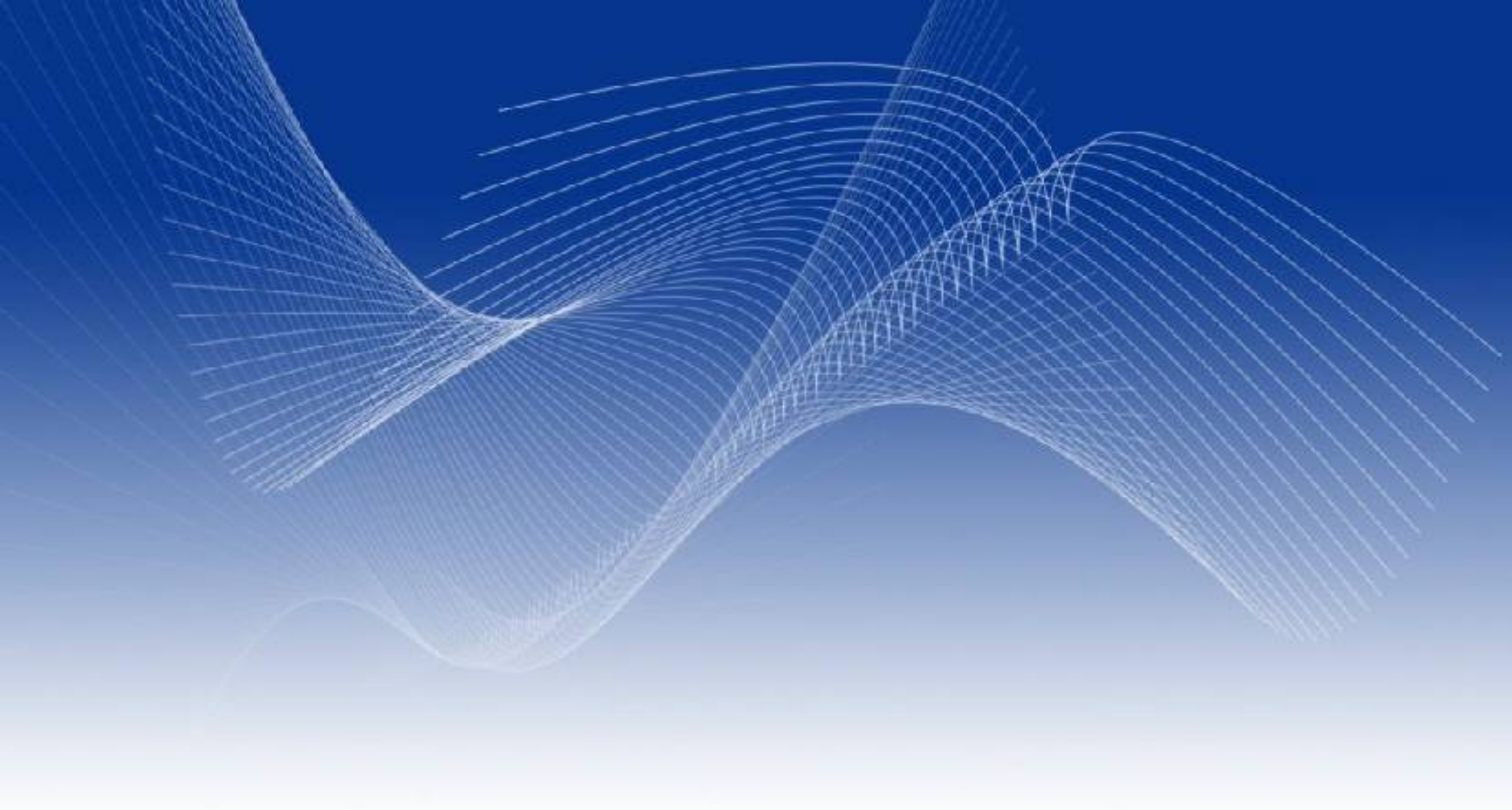


JavaFX on javac: A Case Study

September 25, 2008 **JVM Language Summit**

Tom Ball
Google

Overview

- JavaFX Script Compiler Requirements
- Why Use javac?
- javafxc Design
- Lessons Learned

JavaFX Script Compiler Requirements

- Initial Requirements:
  - Compile .fx scripts to JVM class files
  - Substantially improve performance
  - Substantially reduce footprint

JavaFX Script Compiler Requirements

- Initial Requirements:
  - Compile .fx scripts to JVM class files
  - Substantially improve performance
  - Reduce footprint
- Added Requirements:
  - Specify language

JavaFX Script Compiler Requirements

- Initial Requirements:
  - Compile .fx scripts to JVM class files
  - Substantially improve performance
  - Reduce footprint
- Added Requirements:
  - Specify language
  - IDE support: error reporting, code-completion

JavaFX Script Compiler Requirements

- Initial Requirements:
  - Compile .fx scripts to JVM class files
  - Substantially improve performance
  - Reduce footprint
- Added Requirements:
  - Specify language
  - IDE support: error reporting, code-completion
  - Finish language
    - Expression language changes
    - Animation support
    - Background task support

JavaFX Script Compiler Requirements

- Initial Requirements:
  - Compile .fx scripts to JVM class files
  - Substantially improve performance
  - Reduce footprint
- Added Requirements:
  - Specify language
  - IDE support: error reporting, code-completion
  - Finish language
    - Expression language changes
    - Animation support
    - Background task support
  - Minimum Java version: Java 5

JavaFX Script Compiler Requirements

- Initial Requirements:
  - Compile .fx scripts to JVM class files
  - Substantially improve performance
  - Reduce footprint
- Added Requirements:
  - Specify language
  - IDE support: error reporting, code-completion
  - Finish language
    - Expression language changes
    - Animation support
    - Background task support
  - Minimum Java version: ~~Java 5~~ 1.4.2

JavaFX Script Compiler Requirements

- Initial Requirements:
  - Compile .fx scripts to JVM class files
  - Substantially improve performance
  - Reduce footprint
- Added Requirements:
  - Specify language
  - IDE support: error reporting, code-completion
  - Finish language
    - Expression language changes
    - Animation support
    - Background task support
  - Minimum Java version: ~~Java 5~~ ~~1.4.2~~ Java 6

JavaFX Script Compiler Requirements

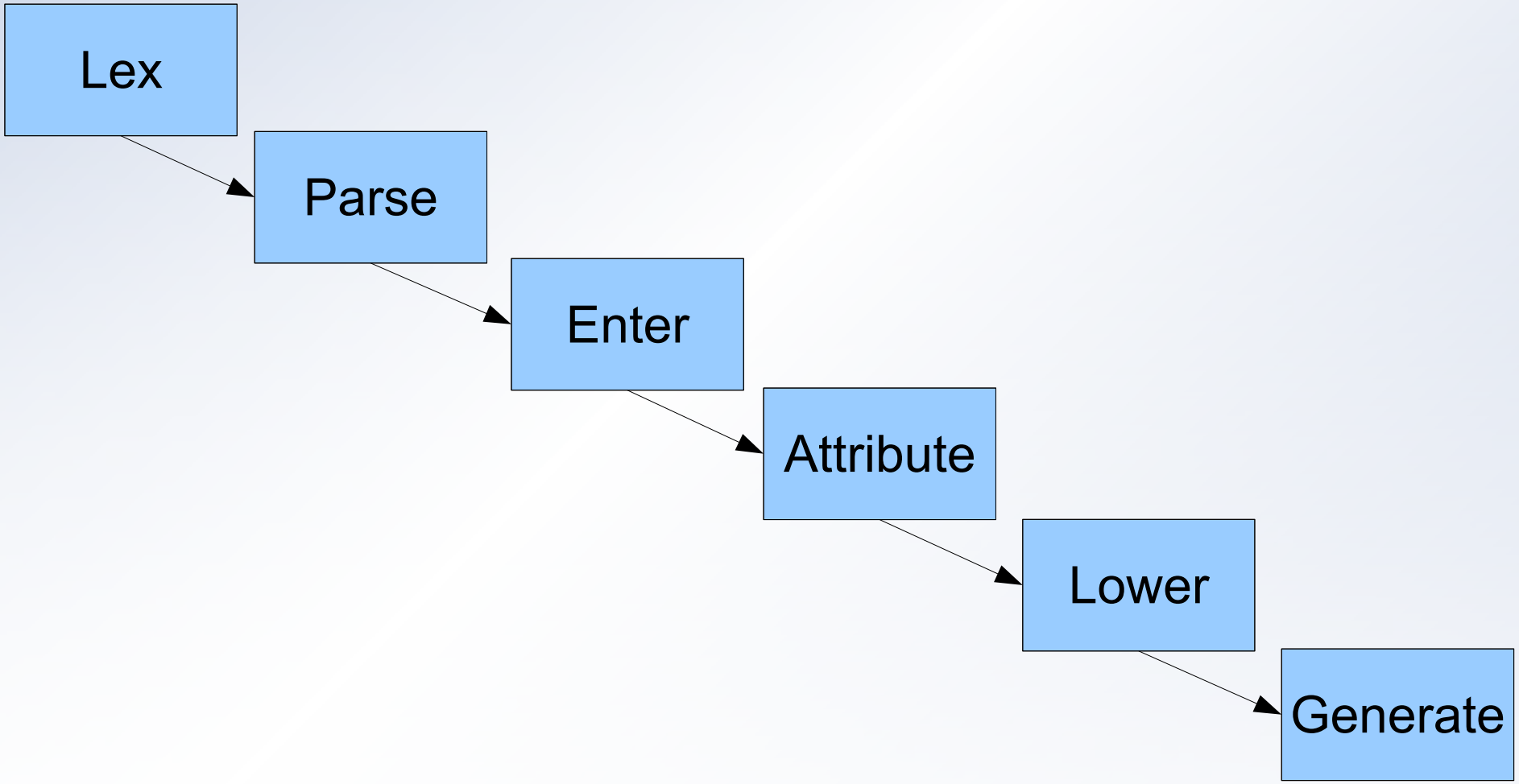
- Initial Requirements:
  - Compile .fx scripts to JVM class files
  - Substantially improve performance
  - Reduce footprint
- Added Requirements:
  - Specify language
  - IDE support: error reporting, code-completion
  - Finish language
    - Expression language changes
    - Animation support
    - Background task support
  - Minimum Java version: ~~Java 5~~ ~~1.4.2~~ ~~Java 6~~ 1.3.1/CLDC

JavaFX Script Compiler Requirements

- Initial Requirements:
  - Compile .fx scripts to JVM class files
  - Substantially improve performance
  - Reduce footprint
- Added Requirements:
  - Specify language
  - IDE support: error reporting, code-completion
  - Finish language
    - Expression language changes
    - Animation support
    - Background task support
  - Minimum Java version: ~~Java 5~~ ~~1.4.2~~ ~~Java 6~~ 1.3.1/CLDC
  - ~~Port~~ Rewrite UI runtime

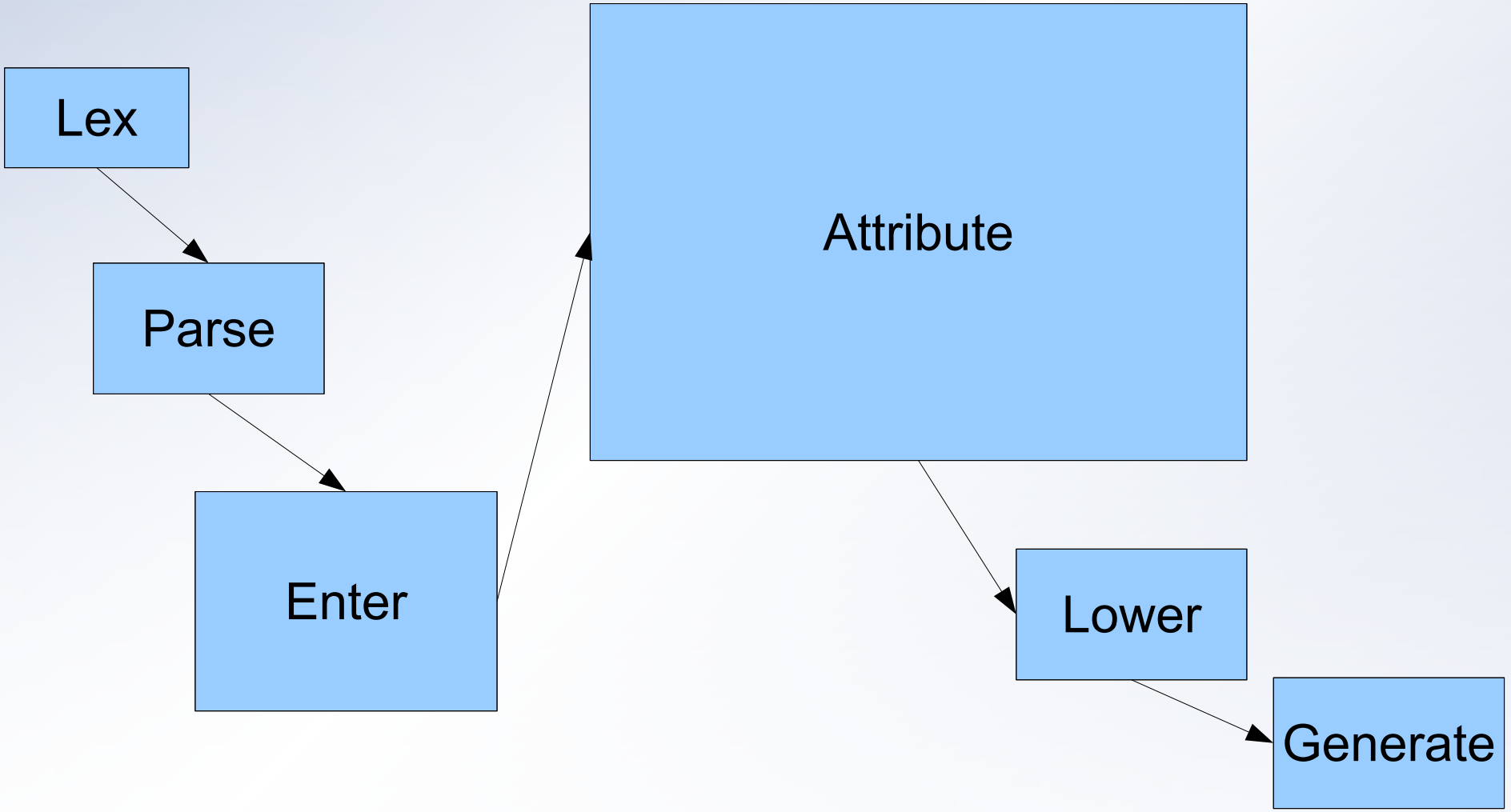
Why Use javac?

Compilers for statically-typed languages are designed as:



Why Use javac?

But over time they become:



Why Use javac?

- Wicked fast
  - Try running it hot
- Rigorous type definition and verification
  - I wanted its TCK
- Re-entrant design
  - Contexts hold service instances
  - Supports IDE editors well
- Error tolerant
  - NetBeans Java editor team work

Embrace and Extend

- All javac phases and singletons live in a Context
- Factories can be “pre-registered” for Context elements
- Subclass phases and singletons to change functionality
- Write your own Main
  - Controls phases and handles options
- Register factories in new Contexts
  - javac code doesn't need to be modified ...
  - ... except for fixing the above when necessary

javafxc Design

- Phases:
  - ANTLR-based parser
    - Great paid support from author
  - JavaFX attribution
  - JavaFX AST -> Java AST translation
  - Java Attribution, Lower phases (unmodified)
  - Generate modified to support block expressions
- JavaFX attribution phase has grown over time
  - Improved error reporting

Lessons Learned

- Get the language description in writing
  - A prototype is not a spec

Lessons Learned

- Get the language description in writing
  - A prototype is not a spec
- javac works best for languages that map to Java
  - As FX moved from Java, javac's benefits faded

Lessons Learned

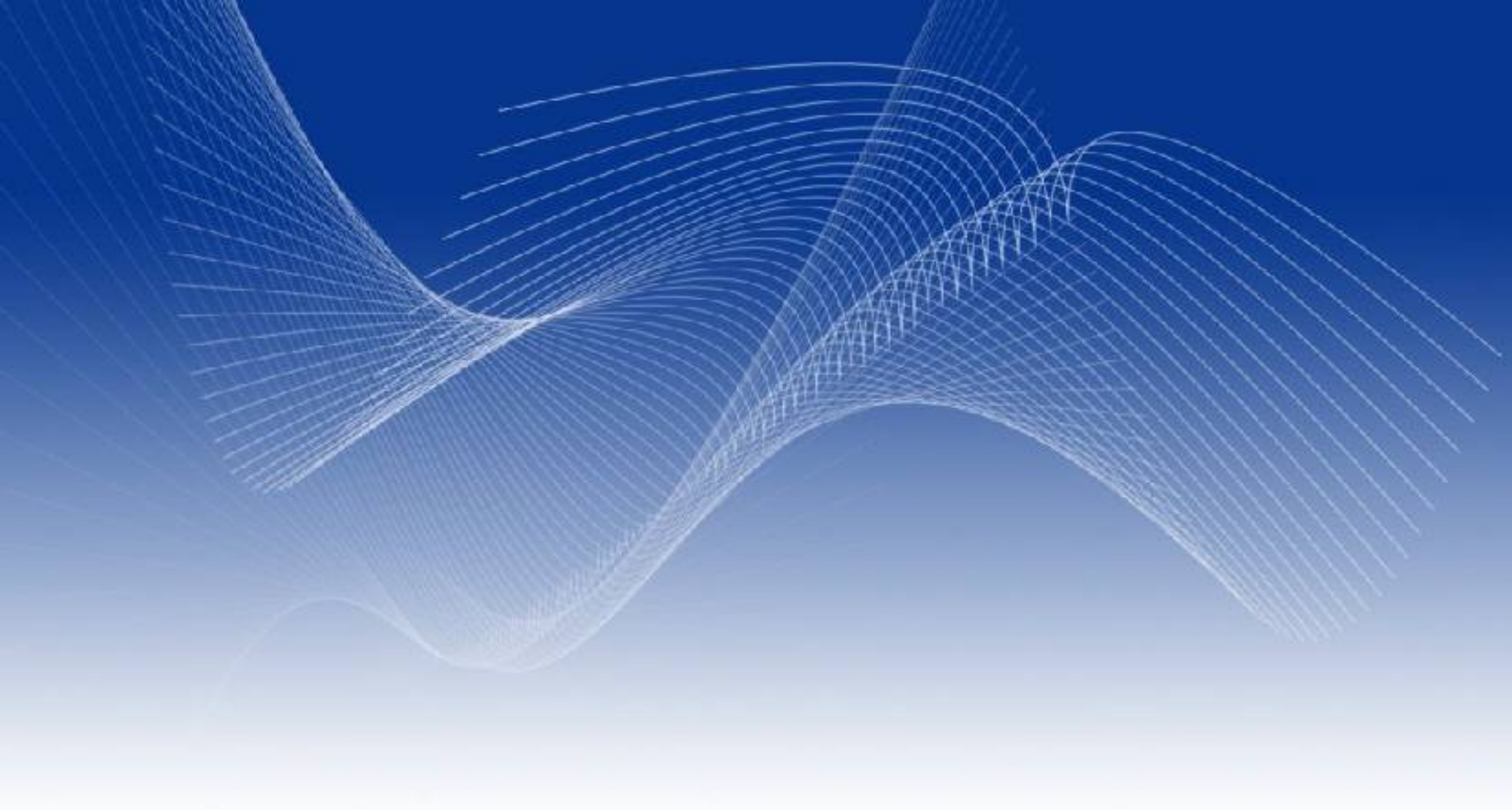
- Get the language description in writing
  - A prototype is not a spec
- javac works best for languages that map to Java
  - As FX moved from Java, javac's benefits faded
- TCK re-use was “slim to none”

Lessons Learned

- Get the language description in writing
  - A prototype is not a spec
- javac works best for languages that map to Java
  - As FX moved from Java, javac's benefits faded
- TCK re-use was “slim to none”
- Watch out for schedule mis-matches
  - Java 6, OpenJDK rollout locked javac source base
  - NetBeans 6 delayed error-recovery patches

Lessons Learned

- Get the language description in writing
  - A prototype is not a spec
- javac works best for languages that map to Java
  - As FX moved from Java, javac's benefits faded
- TCK re-use was “slim to none”
- Watch out for schedule mis-matches
  - Java 6, OpenJDK rollout locked javac source base
  - NetBeans 6 delayed error-recovery patches
- Run from high-visibility projects
  - Languages need time and feedback to gel
  - Execs and computer languages don't mix well



JavaFX on javac: A Case Study

September 25, 2008 **JVM Language Summit**

Tom Ball
Google