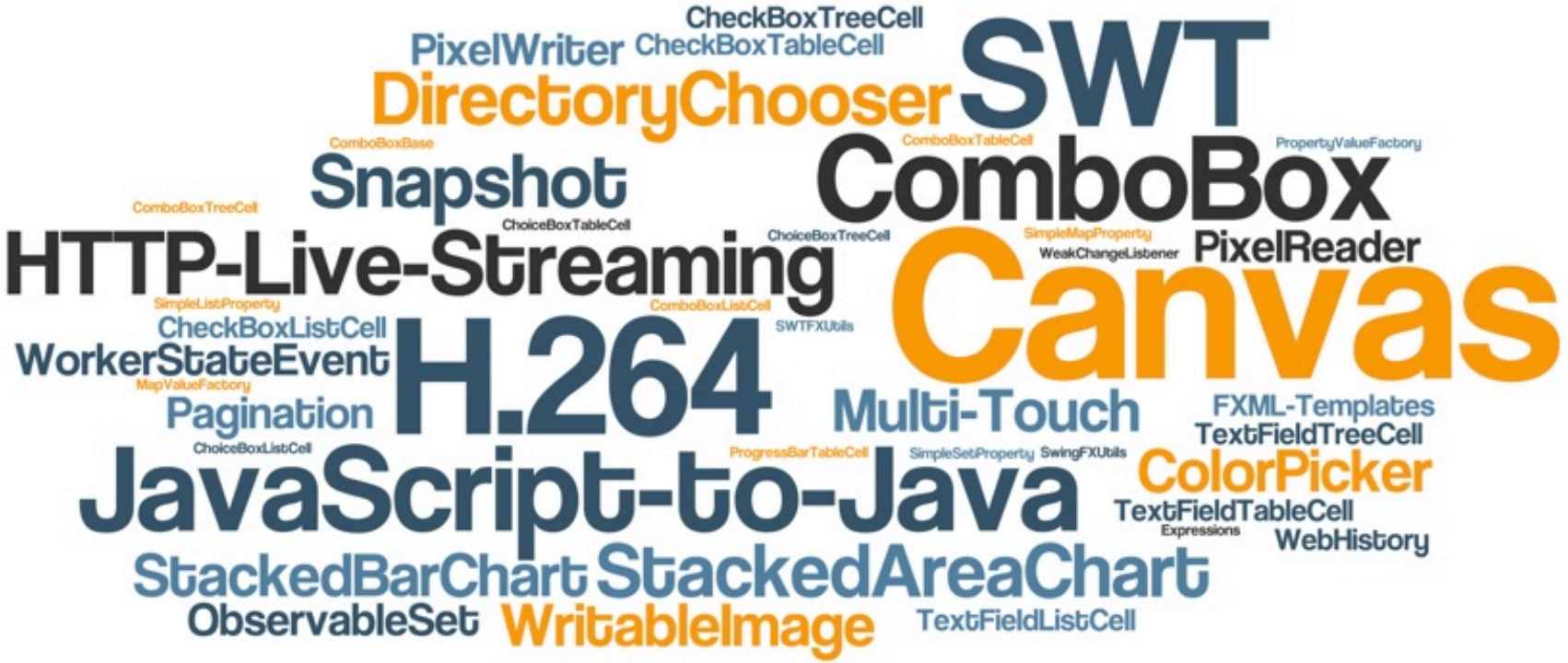

JavaOne™
JavaOne 2012
Technical Keynote
Mark Reinhold (@mreinhold)
Chief Architect, Java Platform Group
MAKE THE
FUTURE
JAVA
ORACLE

Richard Bair
Chief Architect, Java Client Group

Jasper Potts
Developer Experience Architect




```
final SessionFilterCriteria criteria = sessionFilter;
final List<Row> allRows = new ArrayList<Row>(rows);
allRows.addAll(filtered);

filterTask = new Task<Runnable>() {
    @Override protected Runnable call() throws Exception {
        /*...*/
    }
};

filterTask.setOnSucceeded(new EventHandler<WorkerStateEvent>() {
    @Override public void handle(WorkerStateEvent event) {
        final Runnable r = filterTask.getValue();
        if (r != null) r.run();
    }
});

FILTER_EXECUTOR.submit(filterTask);
```



```
// A map of sessions which need to move from being filtered
// to being unfiltered
final Map<Row, List<SessionView>> sessionsToRestore
    = new HashMap<Row, List<SessionView>>();

// A map of sessions which need to move to being filtered
final Map<Row, List<SessionView>> sessionsToFilter
    = new HashMap<Row, List<SessionView>>();
```

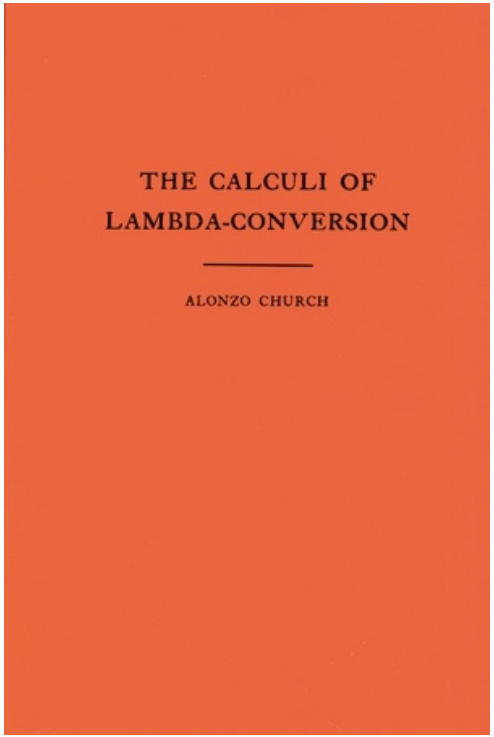
```
private boolean shouldKeep(Session s, SessionCriteria criteria) {
    final String text = criteria.getText();
    final Set<Track> tracks = criteria.getTracks();
    final Set<SessionType> types = criteria.getSessionTypes();

    return (s.getTitle().toLowerCase().contains(text) ||
        s.getSpeakersDisplay().toLowerCase().contains(text) ||
        s.getSummary().toLowerCase().contains(text) ||
        s.getAbbreviation().toLowerCase().contains(text)) &&
        (!(t != null && tracks.contains(t)) &&
        !(st != null && types.contains(st)));
}
```

```
for (Row row : allRows) {
    for (SessionView view : row.sessions) {
        if (isCancelled()) return null;
        if (view.filtered.get() && shouldKeep(view.session, criteria)) {
            List<SessionView> views = sessionsToRestore.get(row);
            if (views == null) {
                views = new ArrayList<SessionView>();
                sessionsToRestore.put(row, views);
            }
            views.add(view);
        } else if (!view.filtered.get() && !shouldKeep(view.session, criteria)) {
            List<SessionView> views = sessionsToRestore.get(row);
            if (views == null) {
                views = new ArrayList<SessionView>();
                sessionsToFilter.put(row, views);
            }
            views.add(view);
        }
    }
}
```


Brian Goetz

Java Language Architect




```
// Event handler for when things go terribly wrong
filterTask.setOnFailed(new EventHandler<WorkerStateEvent>() {
    @Override public void handle(WorkerStateEvent e) {
        e.getSource().getException().printStackTrace();
    }
});
```



```
// Event handler for when things go terribly wrong
filterTask.setOnFailed(new EventHandler<WorkerStateEvent>() {
    @Override public void handle(WorkerStateEvent e) {
        e.getSource().getException().printStackTrace();
    }
});
```

```
filterTask.setOnFailed(
   e -> e.getSource().getException().printStackTrace());
```



```
for (Shape s : shapes) {
    if (s.getColor() == BLUE)
        s.setColor(RED);
}
```

```
for (Shape s : shapes) {
    if (s.getColor() == BLUE)
        s.setColor(RED);
}
```

```
shapes.forEach(s -> {
    if (s.getColor() == BLUE)
        s.setColor(RED);
});
```



```
interface Collection<T> {
    default void forEach(Block<T> action) {
        for (T t : this)
            action.apply(t);
    }
}
```



```
shapes.forEach(s -> {
    if (s.getColor() == BLUE)
        s.setColor(RED);
});
```

```
shapes.forEach(s -> {
    if (s.getColor() == BLUE)
        s.setColor(RED);
});

shapes.filter(s -> s.getColor() == RED)
      .forEach(s -> { s.setColor(BLUE); });
```



```
int sumOfWeight = shapes.parallel()
                        .filter(s -> s.getColor() == BLUE)
                        .map(s -> s.getWeight())
                        .sum();
```



```
for (Row row : allRows) {
    for (SessionView view : row.sessions) {
        if (isCancelled()) return null;
        if (view.filtered.get() && shouldKeep(view.session, criteria)) {
            List<SessionView> views = sessionsToRestore.get(row);
            if (views == null) {
                views = new ArrayList<SessionView>();
                sessionsToRestore.put(row, views);
            }
            views.add(view);
        } else if (!view.filtered.get() && !shouldKeep(view.session, criteria)) {
            List<SessionView> views = sessionsToRestore.get(row);
            if (views == null) {
                views = new ArrayList<SessionView>();
                sessionsToFilter.put(row, views);
            }
            views.add(view);
        }
    }
}
```



```
List<SessionView> views = sessionsToRestore.get(row);
if (views == null) {
    views = new ArrayList<SessionView>();
    sessionsToRestore.put(row, views);
}
views.add(view);
```

```
List<SessionView> views = sessionsToRestore.get(row);
if (views == null) {
    views = new ArrayList<SessionView>();
    sessionsToRestore.put(row, views);
}
views.add(view);
```

```
List<SessionView> views
    = sessionsToRestore.computeIfAbsent(row, () -> new ArrayList<>());
views.add(view);
```



```
List<SessionView> views = sessionsToRestore.get(row);
if (views == null) {
    views = new ArrayList<SessionView>();
    sessionsToRestore.put(row, views);
}
views.add(view);
```

```
List<SessionView> views
    = sessionsToRestore.computeIfAbsent(row, () -> new ArrayList<>());
views.add(view);
```

```
sessionsToRestore.computeIfAbsent(row, () -> new ArrayList<>())
                 .add(view);
```



```
for (Row row : allRows) {
    for (SessionView view : row.sessions) {
        if (isCancelled()) return null;
        if (view.filtered.get() && shouldKeep(view.session, criteria)) {
            List<SessionView> views = sessionsToRestore.get(row);
            if (views == null) {
                views = new ArrayList<SessionView>();
                sessionsToRestore.put(row, views);
            }
            views.add(view);
        } else if (!view.filtered.get() && !shouldKeep(view.session, criteria)) {
            List<SessionView> views = sessionsToRestore.get(row);
            if (views == null) {
                views = new ArrayList<SessionView>();
                sessionsToFilter.put(row, views);
            }
            views.add(view);
        }
    }
}
```



```
Factory<ArrayList<SessionView>> fact = () -> new ArrayList<>();
for (Row row : allRows) {
    row.sessions
       .filter(view -> view.filtered.get()
                       && shouldKeep(view.session, criteria))
       .forEach(view -> sessionsToRestore.computeIfAbsent(row, fact)
                                         .add(view));
    row.sessions
       .filter(view -> !view.filtered.get()
                       && !shouldKeep(view.session, criteria))
       .forEach(view -> sessionsToFilter.computeIfAbsent(row, fact)
                                        .add(view));
}
```

Project Lambda

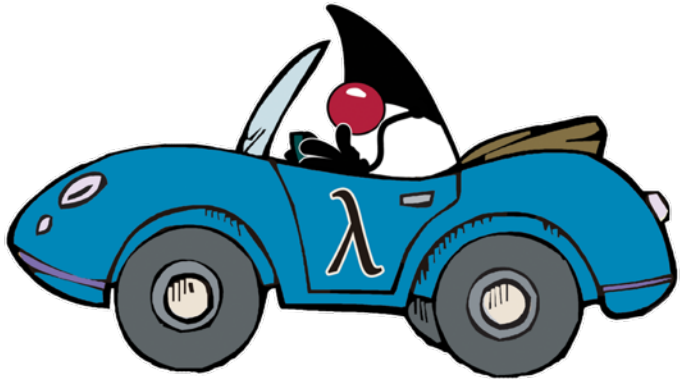
http://openjdk.java.net/projects/lambda

http://jdk8.java.net/lambda

Project Lambda

http://openjdk.java.net/projects/lambda

http://jdk8.java.net/lambda



Project Lambda

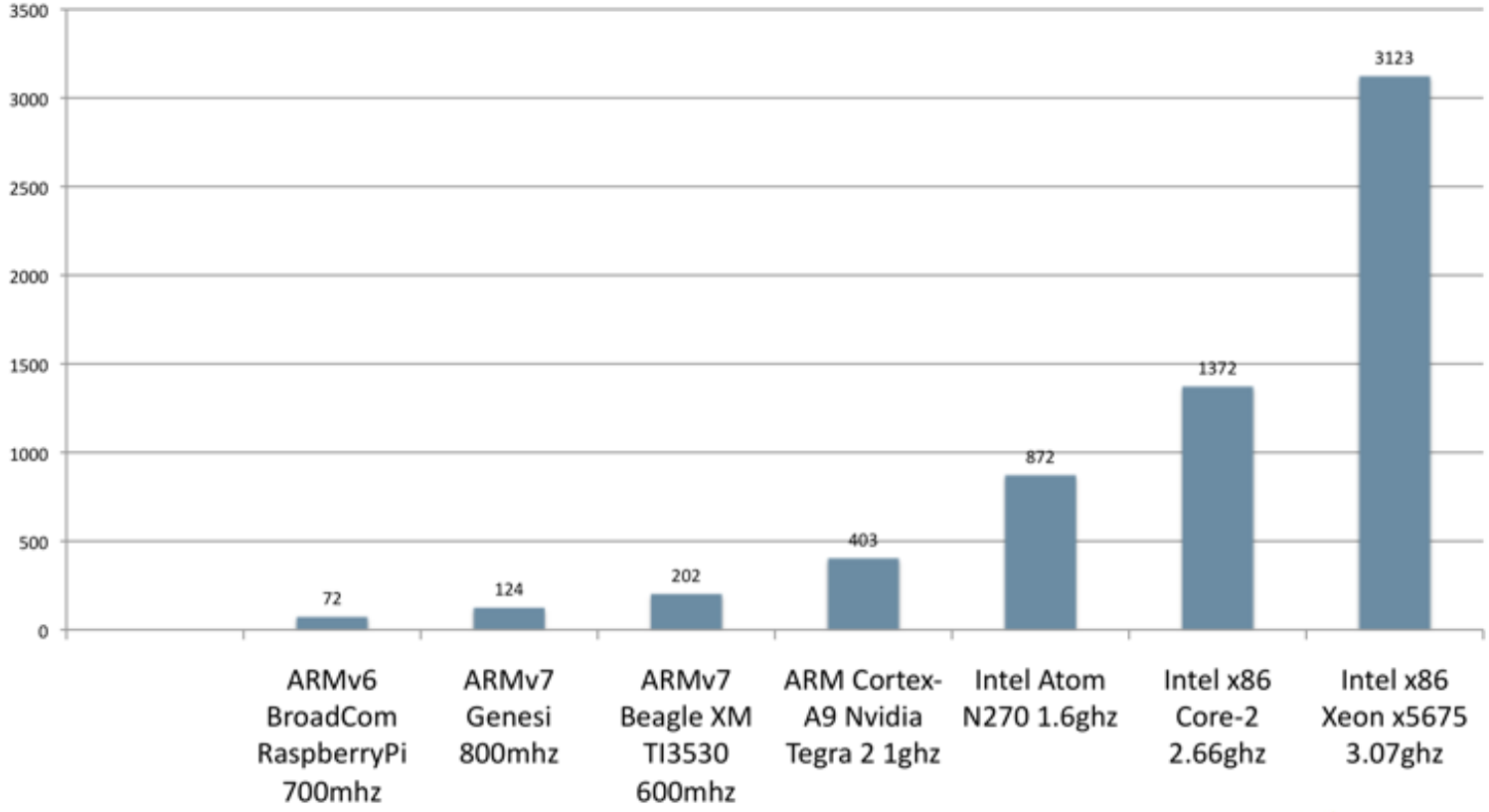
http://openjdk.java.net/projects/lambda

http://jdk8.java.net/lambda

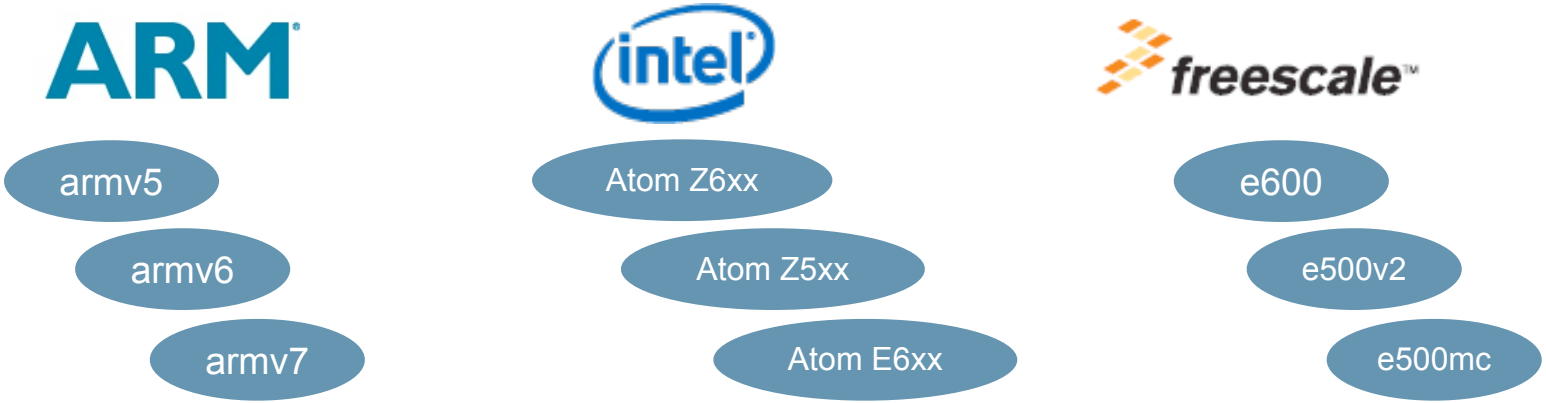
Bob Vandette

Java Embedded Architect

Memory Bandwidth Comparison



Variations in Embedded Architectures



ARM ABI
EABI,HFABI

Big/Little
Endian

Single
Multi-Core

32 vs. 64 bit

soft-float
hard-float

Memory
TSO,RMO

GPU

Software
Rendering

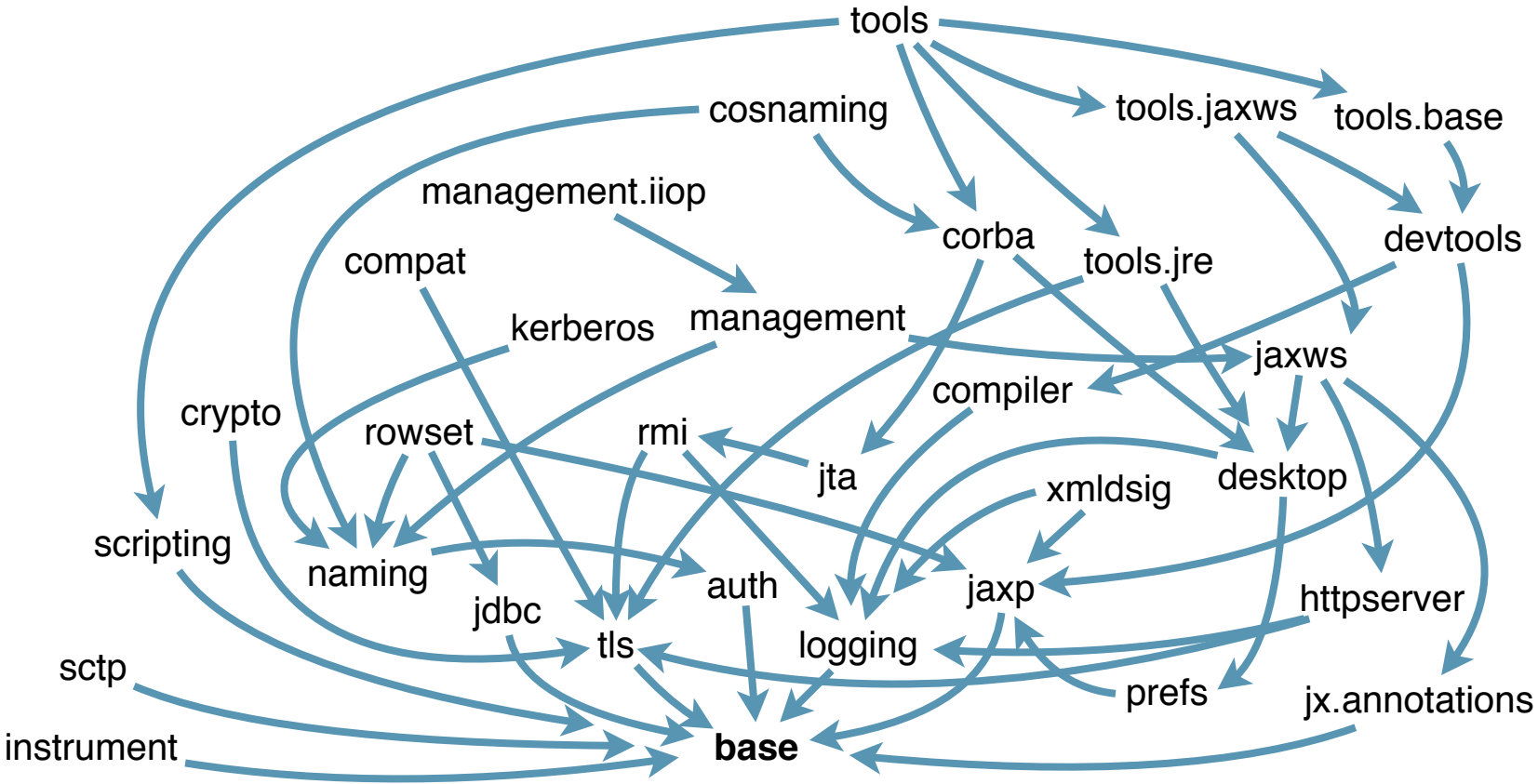
FPU

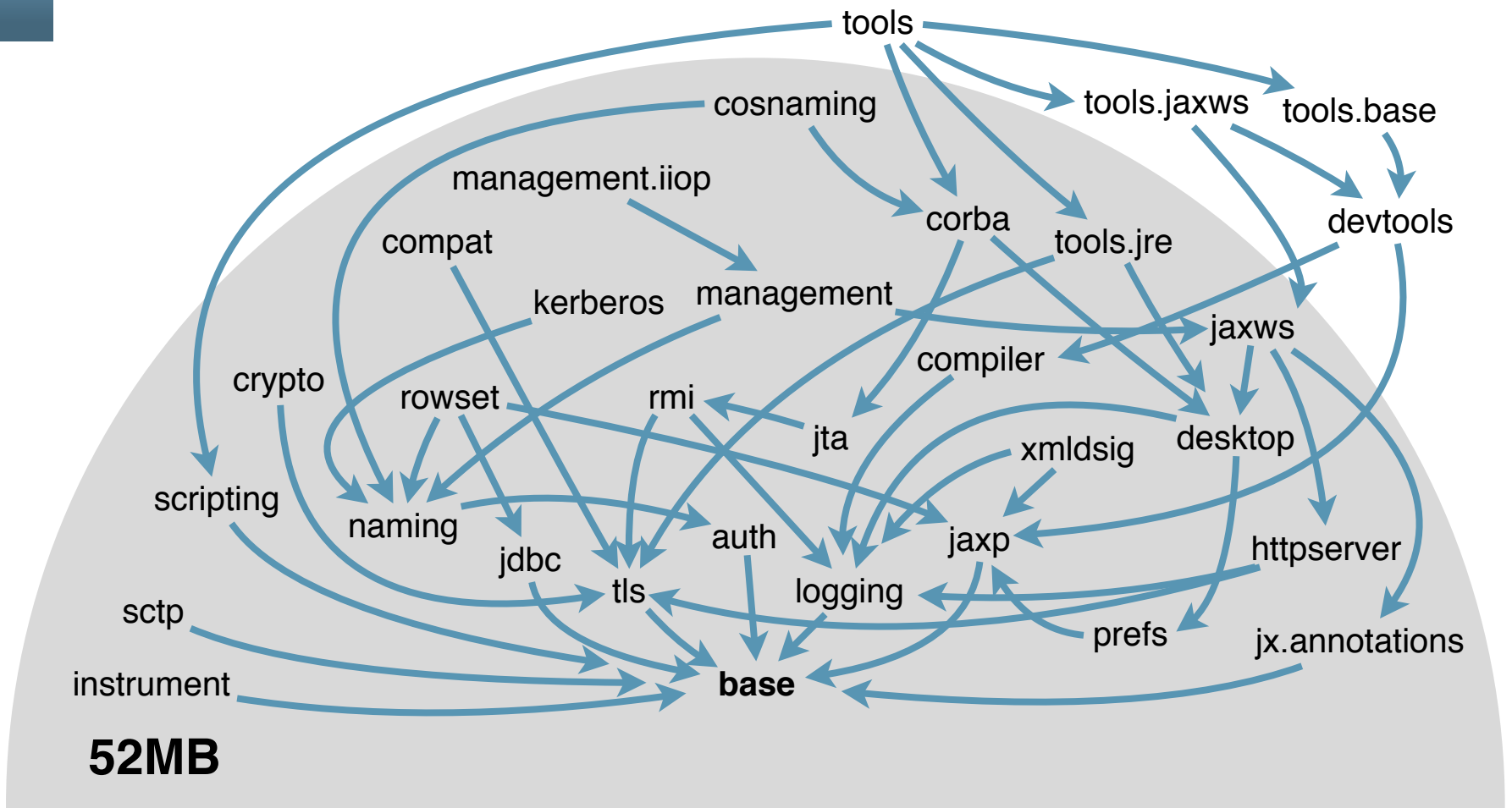
Footprint Metrics

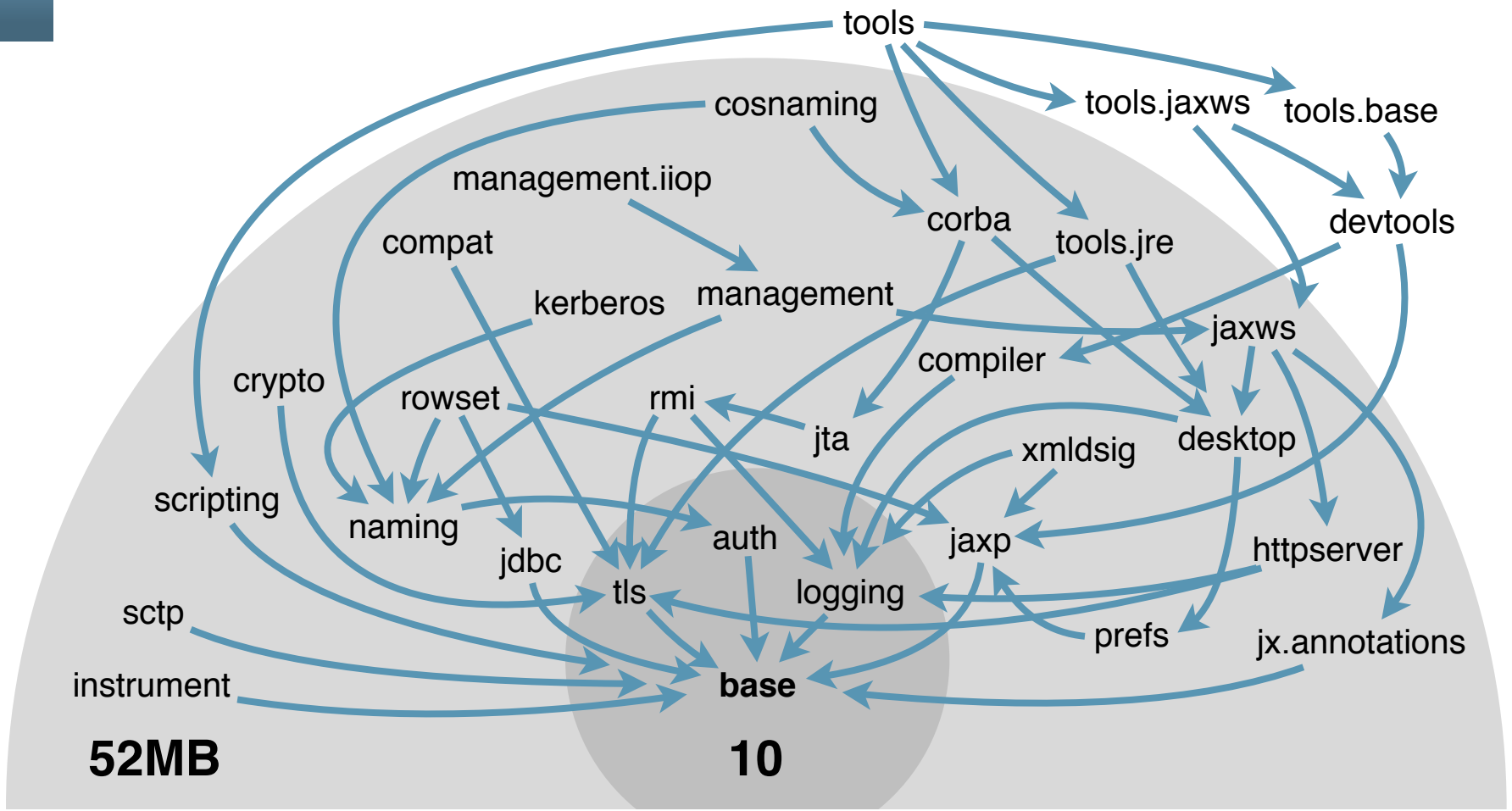
	Current	
Graphics?	No	Yes
Static	43	52
Dynamic	32	64

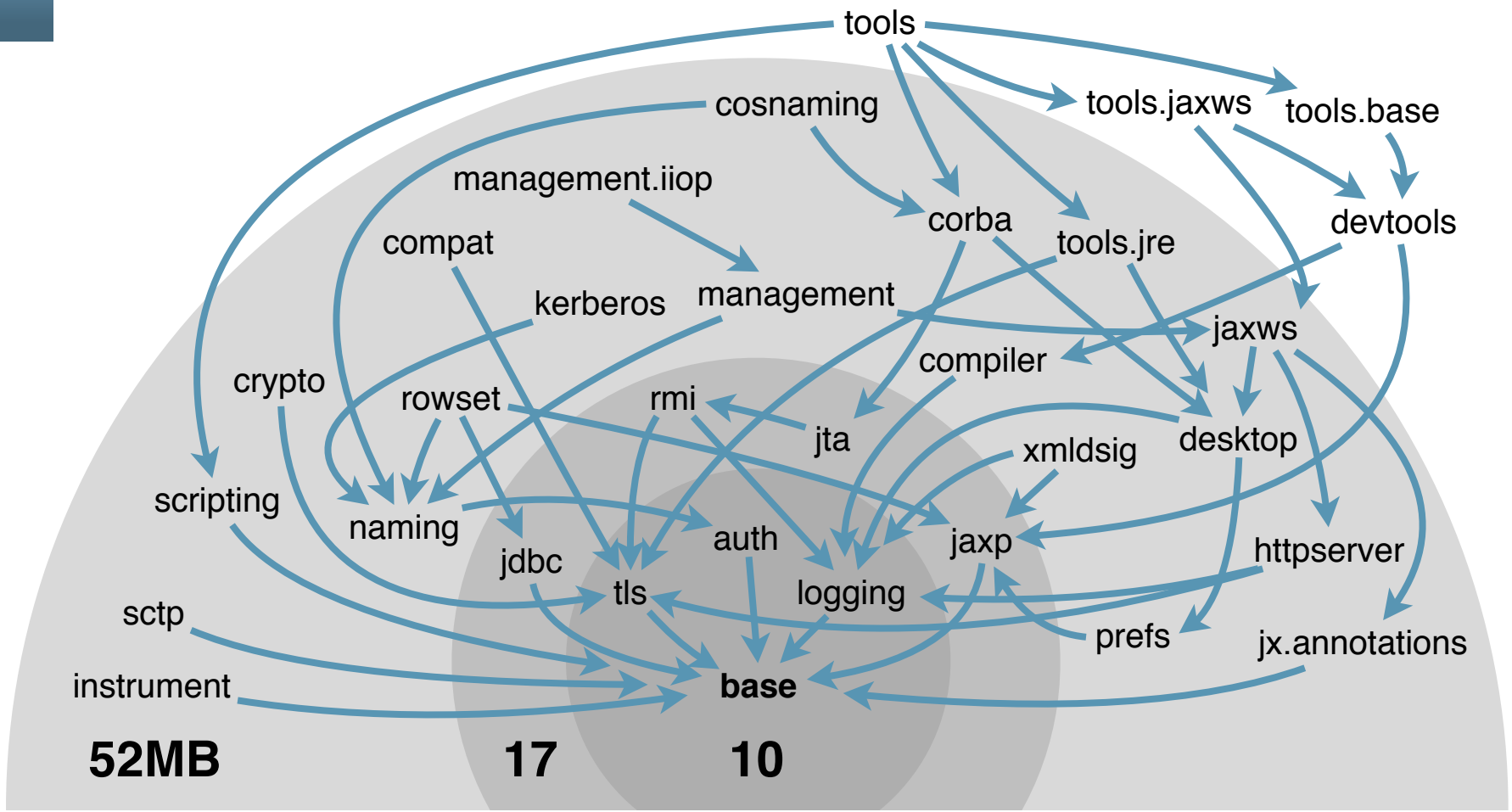
Footprint Metrics

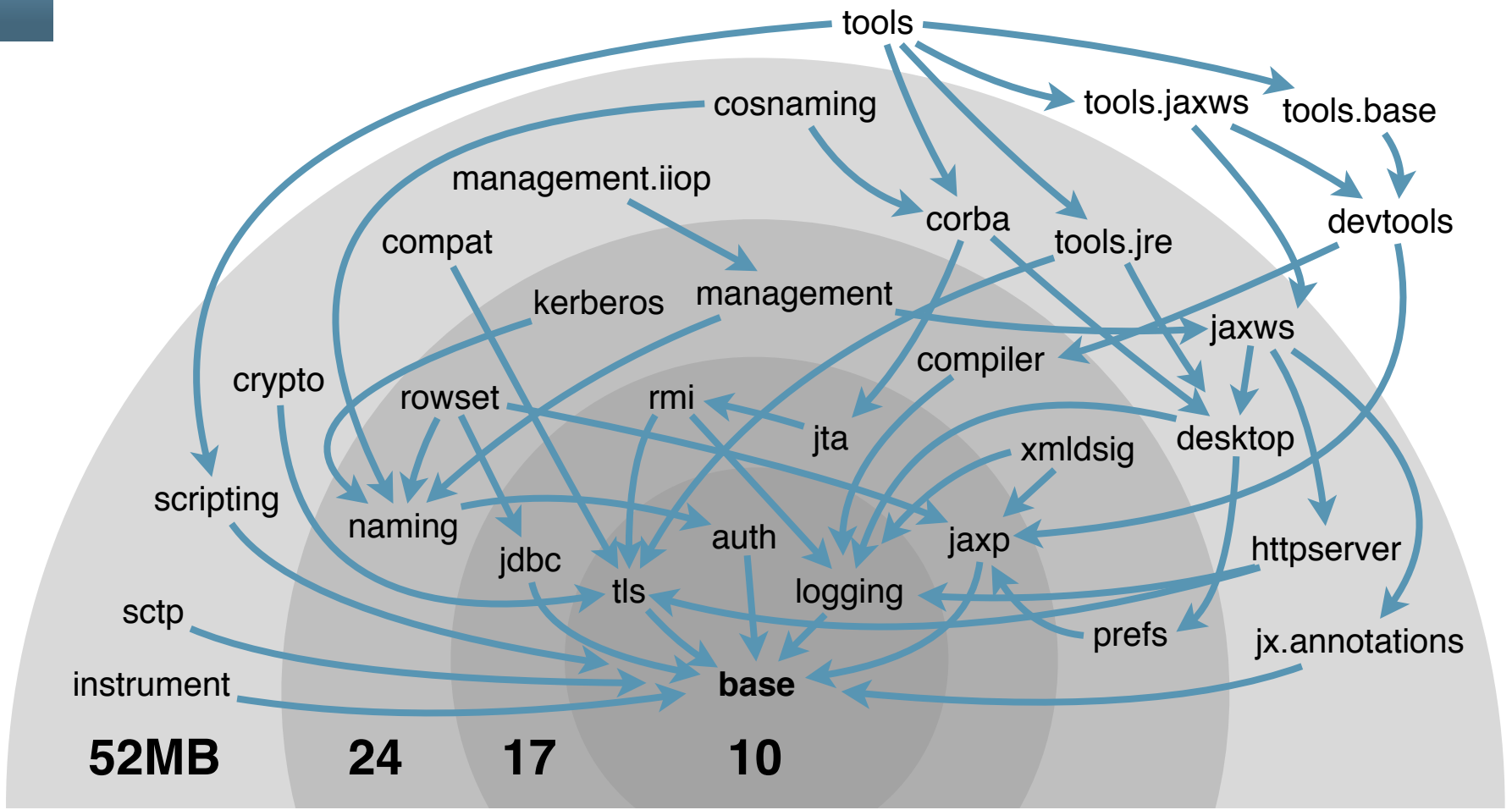
	Current		*Converged (goal)*	
Graphics?	No	Yes	No	Yes
Static	43	52	10	16
Dynamic	32	64	16	32

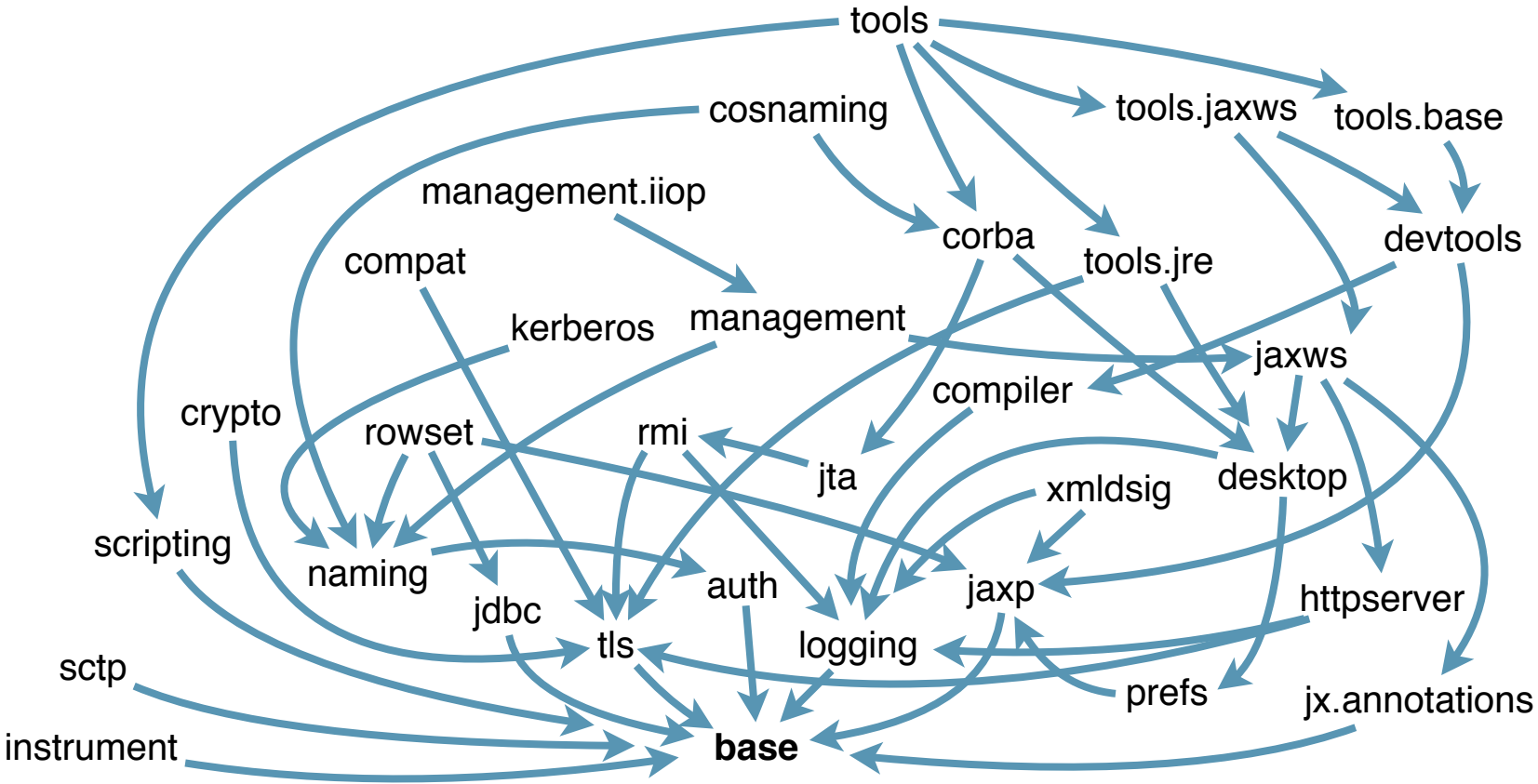

tools
cosnaming
tools.jaxws
tools.base
management.iiop
corba
devtools
compat
tools.jre
kerberos
management
jaxws
compiler
crypto
rowset
rmi
jta
desktop
xmldsig
scripting
naming
auth
jaxp
httpserver
jdbc
tls
logging
sctp
prefs
jx.annotations
instrument
base

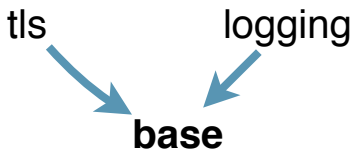

tools
cosnaming
tools.jaxws
tools.base
management.iiop
corba
devtools
compat
tools.jre
kerberos
management
jaxws
compiler
crypto
rowset
rmi
jta
desktop
xmldsig
scripting
naming
auth
jaxp
httpserver
jdbc
tls
logging
sctp
prefs
jx.annotations
instrument
base
52MB


tools
cosnaming
tools.jaxws
tools.base
management.iiop
corba
devtools
compat
tools.jre
kerberos
management
jaxws
compiler
crypto
rowset
rmi
jta
desktop
xmldsig
scripting
naming
auth
jaxp
httpserver
jdbc
tls
logging
sctp
prefs
jx.annotations
instrument
base
52MB
10

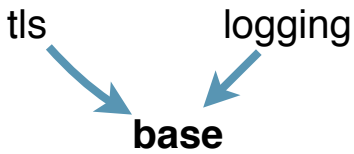

tools
cosnaming
tools.jaxws
tools.base
management.iiop
corba
devtools
compat
tools.jre
kerberos
management
jaxws
compiler
crypto
rowset
rmi
jta
desktop
xmldsig
scripting
naming
auth
jaxp
httpserver
jdbc
tls
logging
sctp
prefs
jx.annotations
instrument
base
52MB
17
10

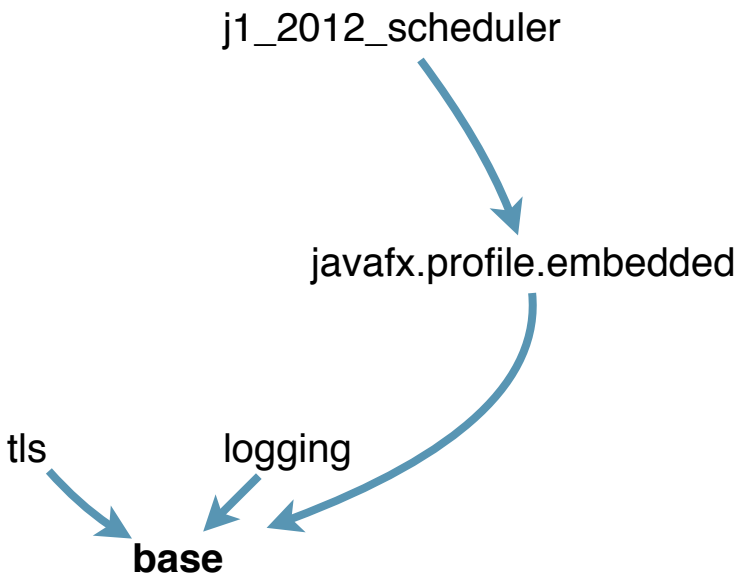

tools
cosnaming
tools.jaxws
tools.base
management.iiop
corba
devtools
compat
tools.jre
kerberos
management
jaxws
compiler
crypto
rowset
rmi
jta
desktop
xmldsig
scripting
naming
auth
jaxp
httpserver
jdbc
tls
logging
sctp
prefs
jx.annotations
instrument
base
52MB
24
17
10

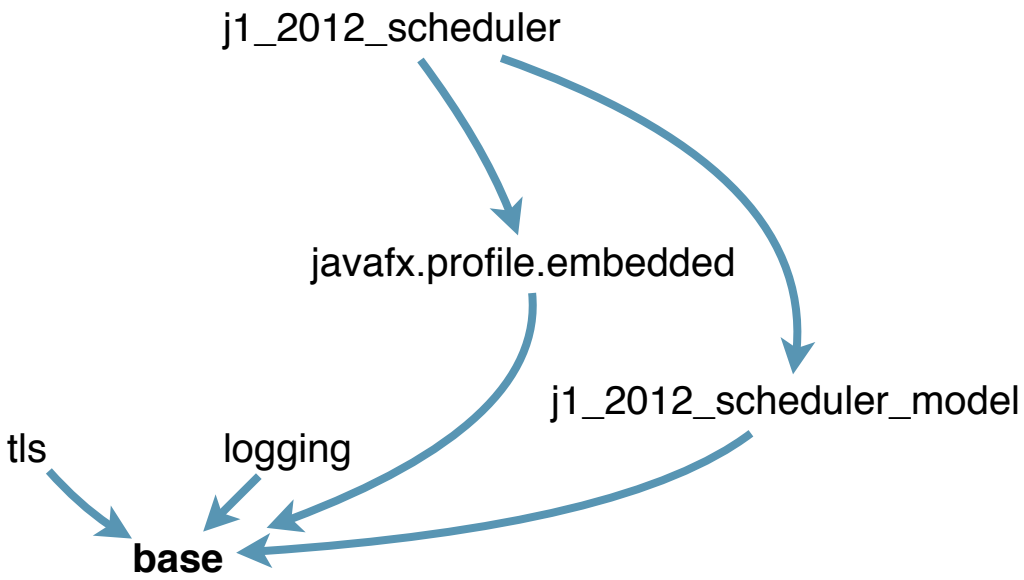

tools
cosnaming
tools.jaxws
tools.base
management.iiop
corba
devtools
compat
tools.jre
kerberos
management
jaxws
compiler
crypto
rowset
rmi
jta
desktop
xmldsig
scripting
naming
auth
jaxp
httpserver
jdbc
tls
logging
sctp
prefs
jx.annotations
instrument
base

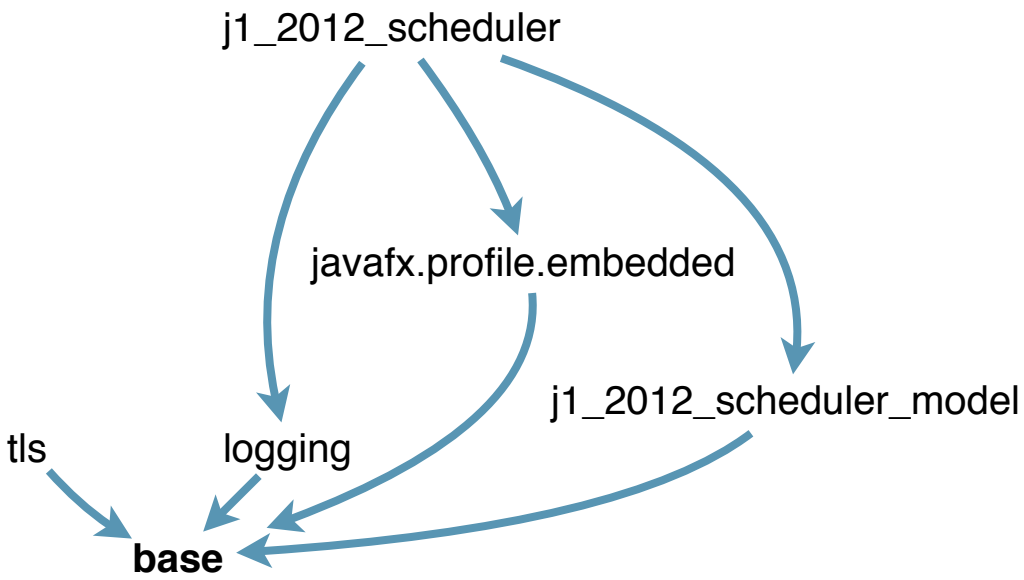

tls
logging
base

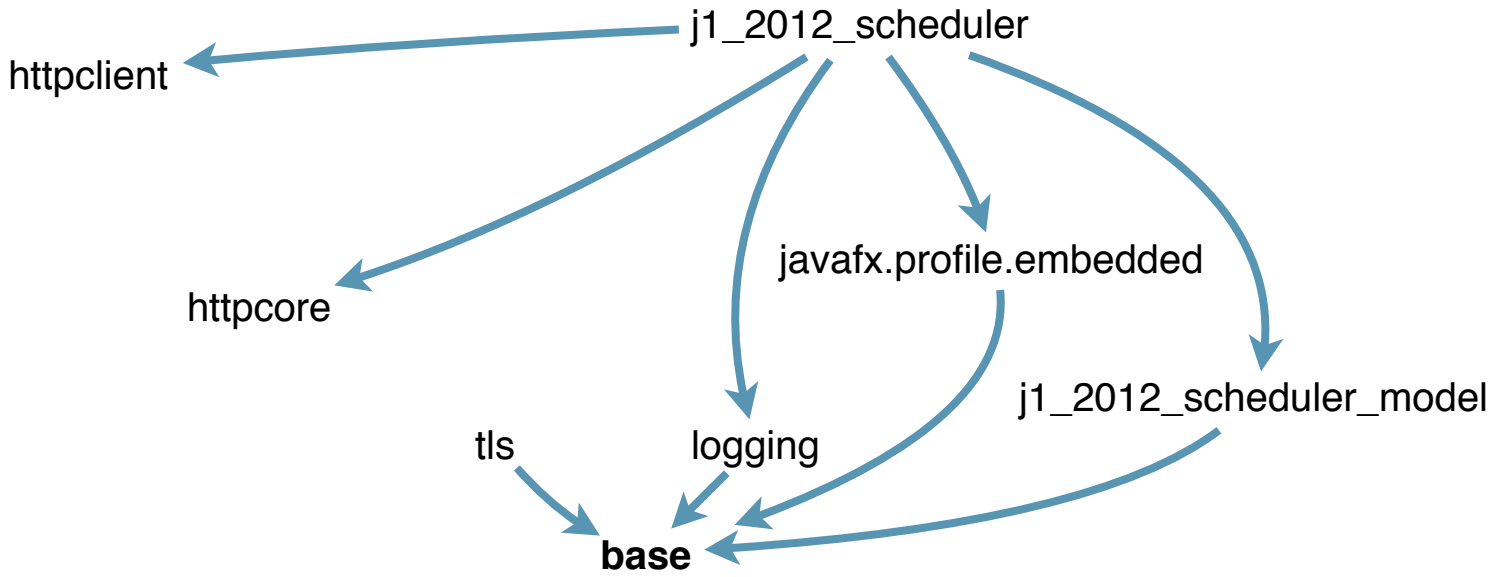
j1_2012_scheduler

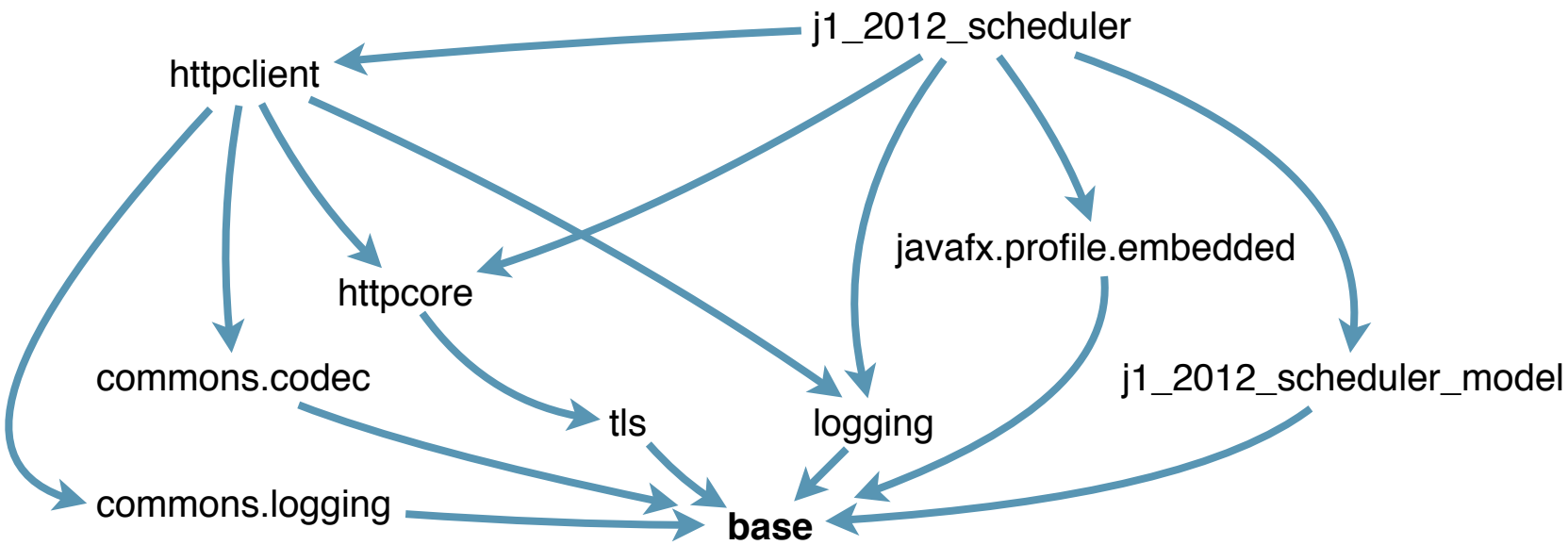



j1_2012_scheduler
javafx.profile.embedded
tls
logging
base


j1_2012_scheduler
javafx.profile.embedded
j1_2012_scheduler_model
tls
logging
base


j1_2012_scheduler
javafx.profile.embedded
j1_2012_scheduler_model
tls
logging
base


j1_2012_scheduler
httpclient
httpcore
javafx.profile.embedded
j1_2012_scheduler_model
tls
logging
base


j1_2012_scheduler
httpclient
javafx.profile.embedded
httpcore
commons.codec
j1_2012_scheduler_model
tls
logging
commons.logging
base

Java 8

Lambda (JSR 335)

Java 8

Lambda (JSR 335)

Java 8

Compact Profiles

Lambda (JSR 335)

Java 8

Compact Profiles

Nashorn

Lambda (JSR 335)

Date/Time API (JSR 310)

Java 8

Compact Profiles

Nashorn

Lambda (JSR 335)

Date/Time API (JSR 310)

Java 8

Compact Profiles

Nashorn

Type Annotations (JSR 308)

NSA Suite B

Base64

HTTP Client

Bulk Data Operations

Unicode 6.2

Prepare for Modularization

Lambda (JSR 335)

Remove the Permanent Generation

Generalized Target-Type Inference

Date/Time API (JSR 310)

Improve Contended Locking

Java 8

Unicode CLDR

Parallel Array Sorting

Compact Profiles

DocTree API

Nashorn

Configurable Secure-Random Number Generation

TLS Server Name Indication

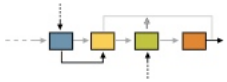
Type Annotations (JSR 308)

Lambda-Form Representation for Method Handles

Java 8

OpenJDK

Java Community Process

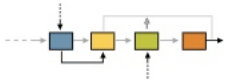


Java 8

JavaOne ORACLE

OpenJDK

Java Community Process



Java 8

http://openjdk.java.net/projects/jdk8
http://openjdk.java.net/projects/jdk8/spec
http://jdk8.java.net

Java 9 … and beyond!

Unified Type System

OpenJFX

Self Tuning JVM

Java 9 … and beyond!

Jigsaw

Ease of use

Reification

Unified Type System

Project Sumatra – Java for GPUs

Improved Native Integration

OpenJFX

Self Tuning JVM

Resource Management

Java 9 … and beyond!

Generic Lang Interoperability

Data Structure Optimizations

Jigsaw

Penrose

Ease of use

More and More Ports

Optimizations

Reification

Multi-Tenancy

The preceding material is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

MAKE THE FUTURE JAVA

ORACLE