# Live Demo: Adapting and Optimizing Java™ ME Applications for Global Deployment

**Allen Lau**
CTO and Co-founder
Tira Wireless
www.tirawireless.com

**Oliver Tabay**
Development Lead,
Content Adaptation

TS-1923

# Goal This Talk

## What You Will Gain

Through examples, learn new techniques and productivity tools to combat mobile deployment challenges

# Agenda

Global Deployment Challenges

Aspect Oriented Programming (AOP)

Live Demo Using Jump

# Agenda

**Global Deployment Challenges**

Aspect Oriented Programming (AOP)

Live Demo Using Jump

# Mobile Devices Are Becoming the Most Prevalent Computing Platform Ever!

| | 2004 | 2009 | Growth |
|---|---|---|---|
| Global Mobile Subscribers | 1.3B | 2.5B | 2x |
| Computing Capable Handsets–Installed Base | 734M | >2.0B | 3x |
| Total Mobile Data Revenue Forecasts | 61B | 189B | 3x |
| Global Mobile Games Revenue Forecasts | 1.5B | 5.6B | 3x |
| Global Mobile Content Revenue Forecasts | 9.5B | 18B | 2x |



Source: OVUM, Sun Microsystems, Strategy Analytics, Informa

# Current Landscape

**Over 180 Mobile Operators have deployed Java based programs– many have custom requirements**

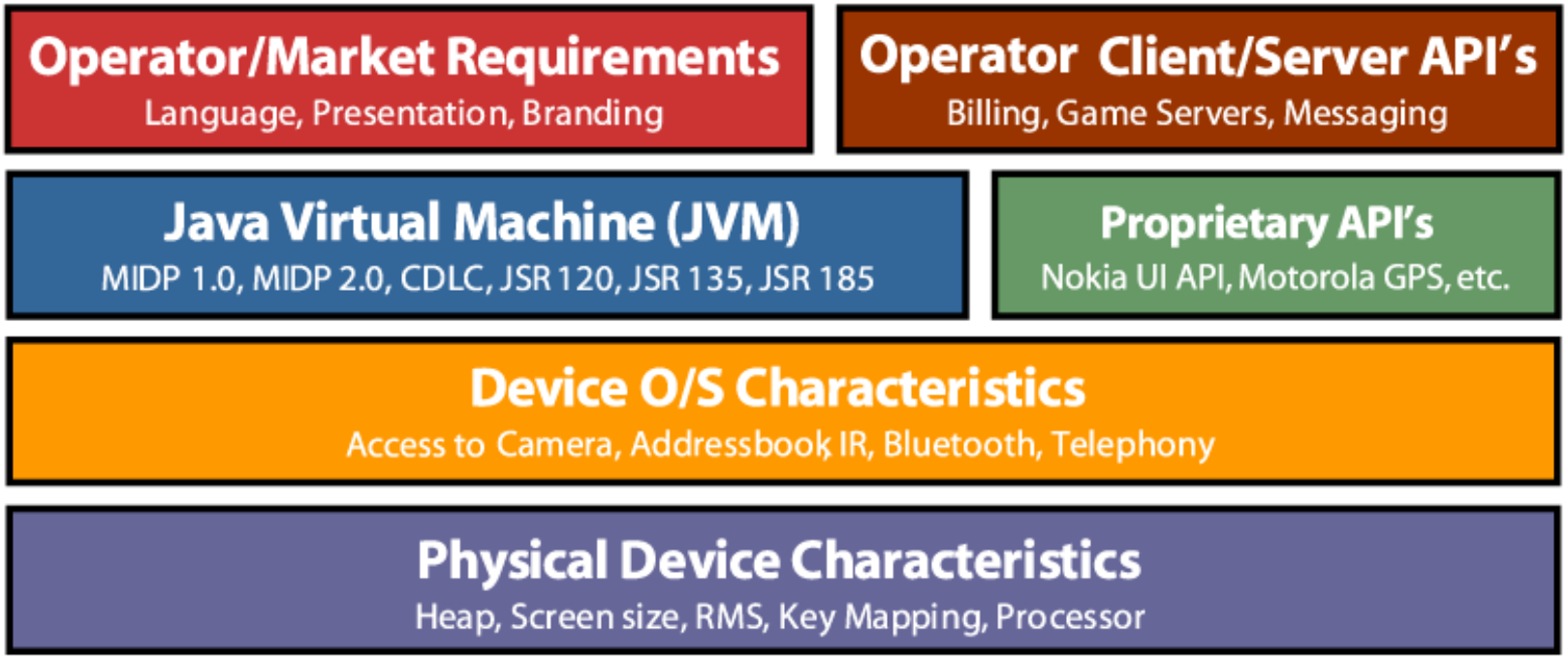

**Over 500 different Java based handset models exist with 20+ new devices being introduced every month**


**It is impossible to deploy the same piece of mobile content on all devices across all channels!**

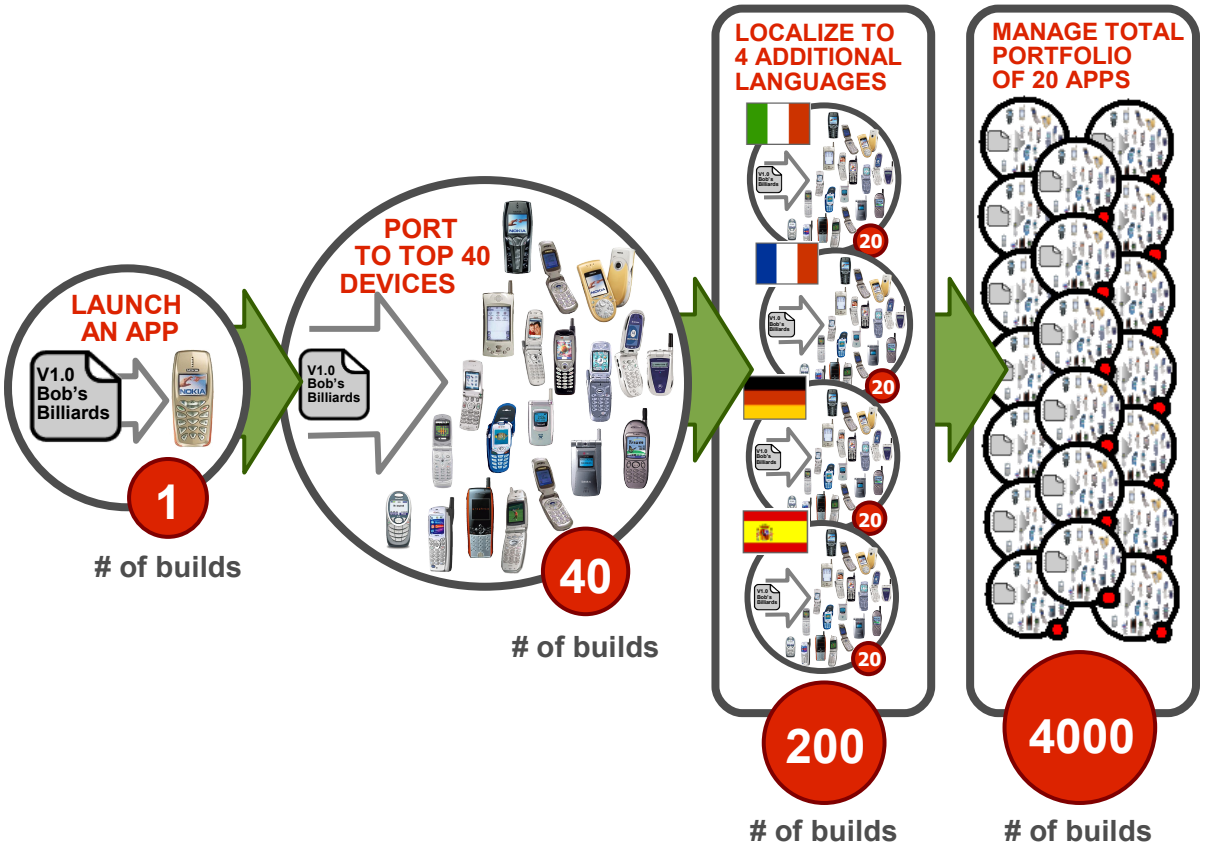
Source: Sun Microsystems, Tira Wireless

# Adaptation and Optimization

## …is the Result of Device and Channel Fragmentation



# Fragmentation Significantly Increases the Need for Multiple Builds…



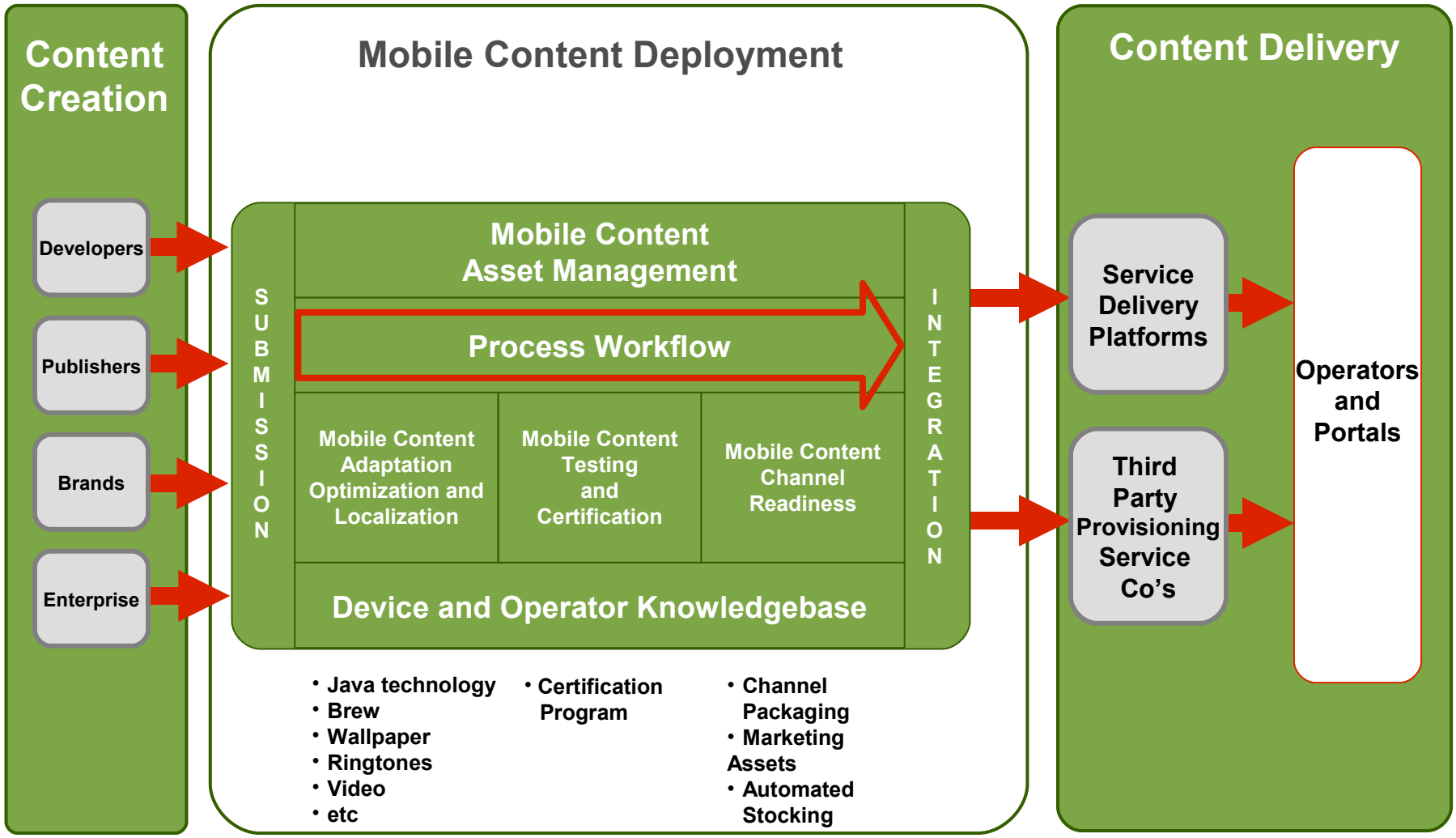

**ADD ONE MORE…**

| | |
|---|---|
| • DEVICE | 100 |
| • APPLICATION | 200 |
| • LANGUAGE | 800 |

NUMBER OF ADDITIONAL BUILDS REQUIRED

**Adding new mobile operators may require duplication of entire portfolio**

# Deployment Complexity



# Example

## Recently Completed Movie-Title-Branded Java ME Game

- Global deployment (Europe, Americas and Asia)
- 127 unique ports supporting over 400 devices in just over 1 month
  - 54 MIDP 1.0 ports
- Packaged builds for different operators and localized builds are excluded
- Similar story for other content types
- New challenges require new processes and techniques

# Agenda

Global Deployment Challenges

**Aspect Oriented Programming (AOP)**

Live Demo Using Jump

# Aspect Oriented Programming

## What is AOP?

- AOP complements OO programming
- Allow developers to dynamically modify static OO models to meet new requirements
- Facilitates modularization of cross-cutting concerns
- This additional code can be kept in a single location rather than having to scatter it across the existing model

# Cross-cutting Concerns

## Is OOP Sufficient in Modularizing Code?

- Most classes in OO performs a single, specific function
- They often share common requirements with other classes, known as cross-cutting concerns
- For example, logging
  - Most likely affects every single logged part of the software
  - Thereby crosscuts all logged classes, methods, and procedures

# Logging Example

```
void paint(Graphics g)
{
    // your paint code
}

void keyPressed(int keyCode)
{
    // your key processing code
}
```

# Logging Example (Cont.)

## Now Adds Logging Code in Old Fashion Way

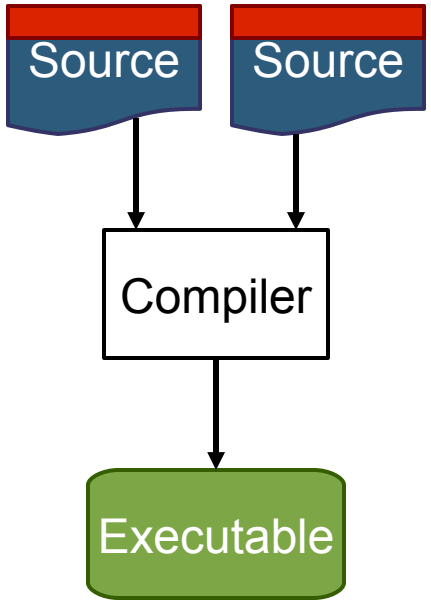
```
void paint(Graphics g)
{
    logging("entering paint");
    // your paint code
    logging("leaving paint");
}

void keyPressed(int keyCode)
{
    logging("entering keyPressed");
    // your key processing code
    logging("entering keyPressed");
}
```

# Logging Example (Cont.)

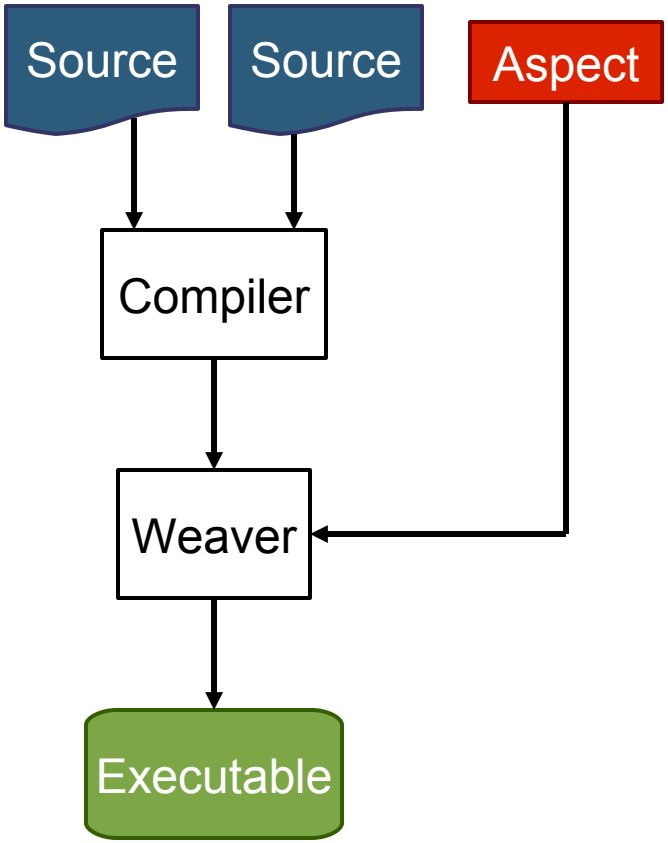
## Pseudo Code of an Aspect

```
loggingAspect
{
   loggableCalls = paint, keyPressed;

   before: loggableCalls
   {
      logging("entering " + $methodName);
   }

   after: loggableCalls
   {
      logging("leaving " + $methodName);
   }
}
```

# Without AOP



# With AOP



# Some New Terminologies

- Pointcut
- Advice
- Aspect

# Pointcut

## The Point of Execution in the Application at Which Cross-cutting Concern Needs to be Applied

```
loggingAspect
{
   loggableCalls = paint, keyPressed;

   before: loggableCalls
   {
      logging("entering " + $methodName);
   }

   after: loggableCalls
   {
      logging("leaving " + $methodName);
   }
}
```

# Advice

## The Code That You Want to Apply to Your Existing Model

```
loggingAspect
{
    loggableCalls = paint, keyPressed;

    before: loggableCalls
    {
        logging("entering " + $methodName);
    }

    after: loggableCalls
    {
        logging("leaving " + $methodName);
    }
}
```

# Aspect

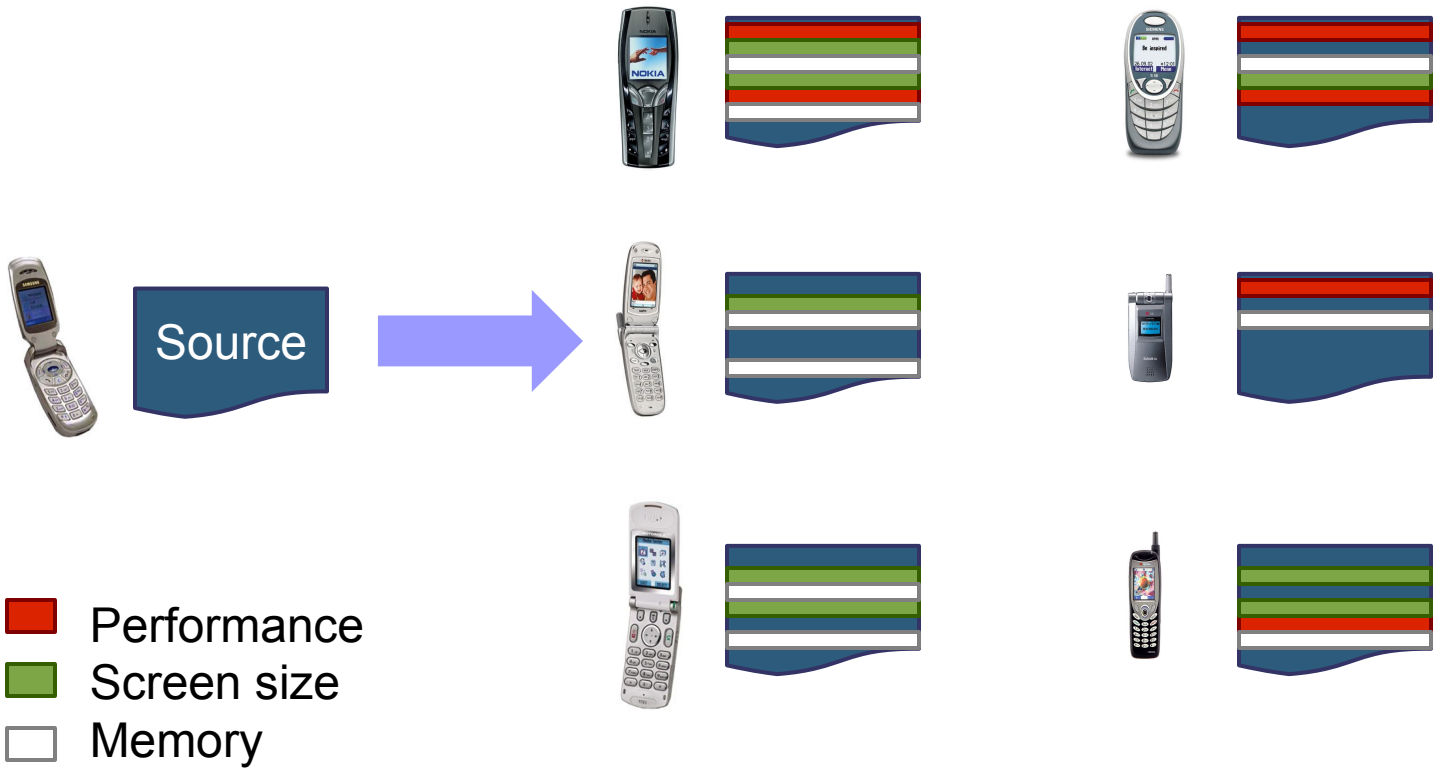
## The Combination of the Pointcut and the Advice

```
loggingAspect
{
   loggableCalls = paint, keyPressed;

   before: loggableCalls
   {
      logging("entering " + $methodName);
   }

   after: loggableCalls
   {
      logging("leaving " + $methodName);
   }
}
```

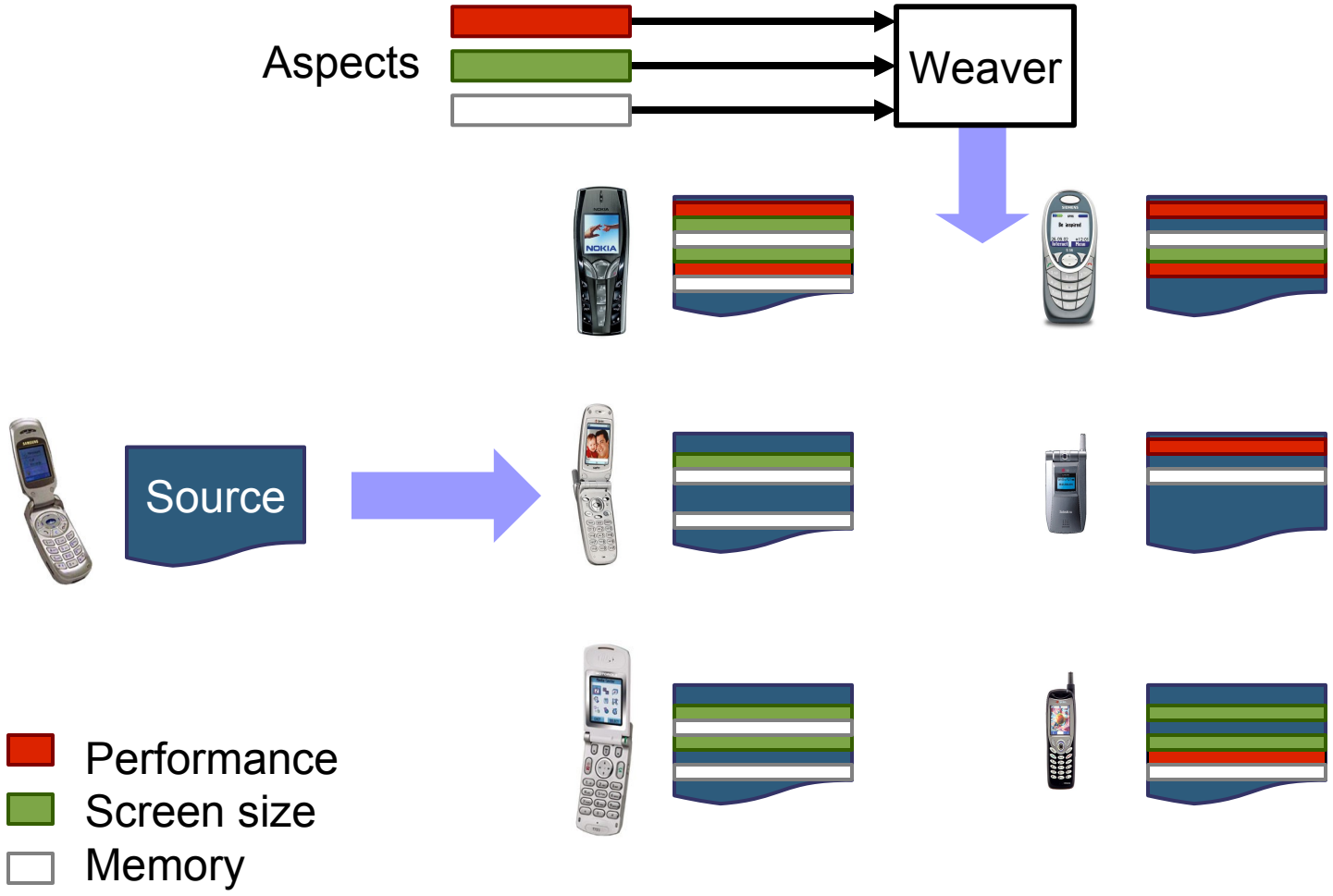
# How AOP Can Help Mobile Deployment?

- Modularizing concerns into aspects
- Reusing aspects for more than one devices
- Reusing aspects for multiple applications

# Without Modularising Concerns Into Aspects



# Modularising and Reusing Concerns Into Aspects



# Popular AOP Implementations

- AspectJ
- JBoss AOP
- AspectC++
- Spring AOP
- Jump Adjustment Library (JAL)

# JAL Pointcut Examples

- IClassPointcut
- IConstructorPointcut
- IExceptionHandlerPointcut
- IFieldAccessPointcut
- IFieldPointcut
- IFieldWriterPointcut
- IMethodCallPointcut
- IMethodPointcut
- INewExpressionPointcut
- IStaticInitializerPointcut

# JAL Advice Examples

- IMethodCallPointcut
  - insertBefore
  - insertAfter
  - replace
  - stub
- IMethodPointcut
  - incrementParameter
  - decrementParameter
  - modifyParameter
  - addCatch
- IClassPointcut
  - addField
  - addMethod

# Example

```
public void moveAllImagesToTheLeft(IAdjustment adj)
{
    IMethodCallPointcut mcpc;

    // get pointcuts to all g.drawImage(img, x, y, anchor)
    mcpc = adj.getMethodCallPointcut(
        "com.tirawireless.game.BaseJumper",
        "paint",
        "drawImage");

    mcpc.decrementParameter(2, 5);
}
```

# Other Alternatives

- Netbeans™ Mobility Pack and other pre-processor solutions
  - Comment-based Pre-processor solution

```
/*#BigDevice#*/
entry = new
TextField("Name:",null,25,TextField.ANY);
/*$BigDevice$*/
/*!#BigDevice#*/
entry = new
TextField("Name:",null,10,TextField.ANY);
/*!#BigDevice#*/
```

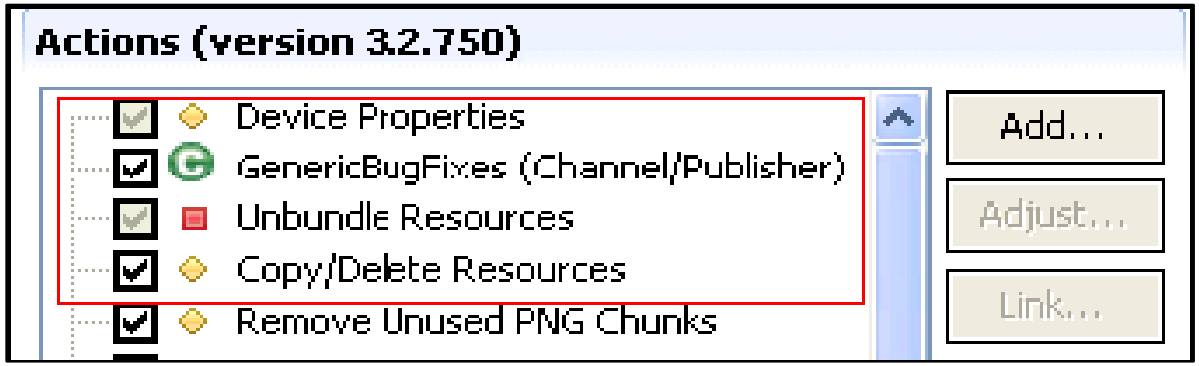
- Intermediate language/abstraction layer
- Duplicate source code

# Demo Agenda

- Reference device
  - Nokia Series 60
- Target devices
  - Motorola V300 series
  - Sony Ericsson S700i
- Logging, profiling and adding new functionality for a specific build
- Reuse aspects through “conditions”

# Introduction to Jump Configurations and Actions

- A “configuration” is the “script” of a particular port
- It consists of a list of actions




- The list tells Jump what transformations to perform on the application
- The actions are executed sequentially
- Target build is created as a result

# DEMO

# Summary

- Mobile content deployment is Much More complex than most people think
- Domain expertise is important
- New challenges require new techniques
- Getting help from mobile deployment specific software is a must

# For More Information

- www.tirawireless.com
- www.tiradeveloper.com
- www.eclipse.org/aspectj
- www.netbeans.org/products/mobility
- wikipedia.org/wiki/aspect-oriented_programming

# Q&A

Allen Lau
Oliver Tabay
Tira Wireless

# Live Demo: Adapting and Optimizing Java™ ME Applications for Global Deployment

**Allen Lau**
CTO and Co-founder
Tira Wireless
www.tirawireless.com

**Oliver Tabay**
Development Lead,
Content Adaptation

TS-1923