
# Running a Foundation to Contain Your Code

David Wood, EVP Research, Symbian Ltd

# The Symbian Platform

*Now becoming Open Source via the Symbian Foundation*

- Symbian OS is the market leading Smartphone OS
- Mature and most widely deployed platform
- Competitive products in mid- and high-end devices
- Operators and developers embrace Symbian OS

7 device manufacturers

> **250 million devices**, > 250 device models

> 250 operators

Tens of thousands of apps, 4 million developers

# The (approximate) scale of the solution

40 million lines of code (device/tools/test/PC)

450,000 source code files

45,000 directories of source code files

2000 software components

97 packages

12 technology domains

1 mobile operating system

FORTUNE

MOBILITY TECH

# The New Smartphones

We take four of the best for a ride. **By Jessi Hempel**

APPLE'S IPHONE MAY HAVE set the standard for a new generation of smartphones, but plenty of competitors, including RIM, Nokia, and Google, are vying to upstage Steve Jobs in the competition for consumers' hearts. RIM's BlackBerry models are the favorite choice for U.S. corporate users, while Nokia's phones hold the lead in the rest of the world. Here are four top picks for the PDA-obsessed.


## RIM BlackBerry Bold

**UPSIDE** A brilliant screen, the best e-mail keyboard available, and a slick operating system.

**DOWNSIDE** It's a bit bulky, and web browsing is still more awkward than on its competitors.

***$299; AT&T***

## Apple iPhone 3G

**UPSIDE** Can handle Microsoft Exchange e-mail and offers lots of fun, downloadable applications.

**DOWNSIDE** The battery runs down quickly, and the keyboard isn't for the typo-prone.

***$199; AT&T***


## Nokia E 71

**UPSIDE** Sleek frame, a 3.2-megapixel camera, and you can use it with any carrier. A toggle switch flips between screens for work and home.

**DOWNSIDE** Requires special headphone jack, and the screen is tiny.

***$465; T-Mobile and AT&T***


## Google Android G1

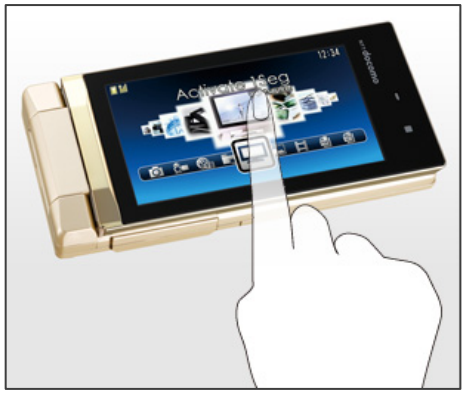
**UPSIDE** Easy to navigate, with a force-quit feature that lets you close a frozen app without rebooting.

**DOWNSIDE** It's chunky and heavy, and the keyboard is, for some, challenging to use.

***$179.99; T-Mobile***

**OUR PICK** Despite a small screen, the Nokia E71 offers a sleek and satisfying experience and stands out from the pack.

# New phones shipped during November 2008



DoCoMo Prime F-01A by Fujitsu

DoCoMo Prime SH-01A by Sharp

LG KT615


Nokia 5800 Express Music

Samsung I 7110

more…

symbian

# Huge mobile opportunities

The mobile devices of the near future will be much more powerful and much more useful than even the best mobile devices of today

enchanting consumer experiences

extra senses: location, motion, eyes, ears…

devices that make **all users** smarter

+make society smarter

extra memory

rich mobile gateway to the digital world

entertainment

business

health

education

journalism

symbian

# Significant mobile challenges

Delays in phone development projects: quality suffers

Complexity

Challenges for developers

Hard for users to find functionality

Small screens, small keyboards

Applications difficult to use

Applications fail to run well when moved to different device or to different network

Larger data causes processing delays

Batteries drain too quickly

Security and privacy issues

# Solving mobile challenges

Deep challenges and rich opportunities need very many developers working on them

Openness:
Create/Embrace an **ecosystem**

Symbian Foundation

Open Source:
Not just innovation
but **deep** community innovation

# The claims of Open Source Software

- Faster time to market
- More eyeballs looking at code, hence higher quality
- Lower barriers to entry, hence greater innovation
- Lower barriers to collaboration
- Lower license fee
- The tide of history…

***And specifically, about Linux:***

- Better APIs?
- A larger ecosystem – including better tools

But: Many mobile phone companies lost huge amounts of money pursuing open source solutions

Or is it just a question of timing?

Over

# Predicting the success of open source

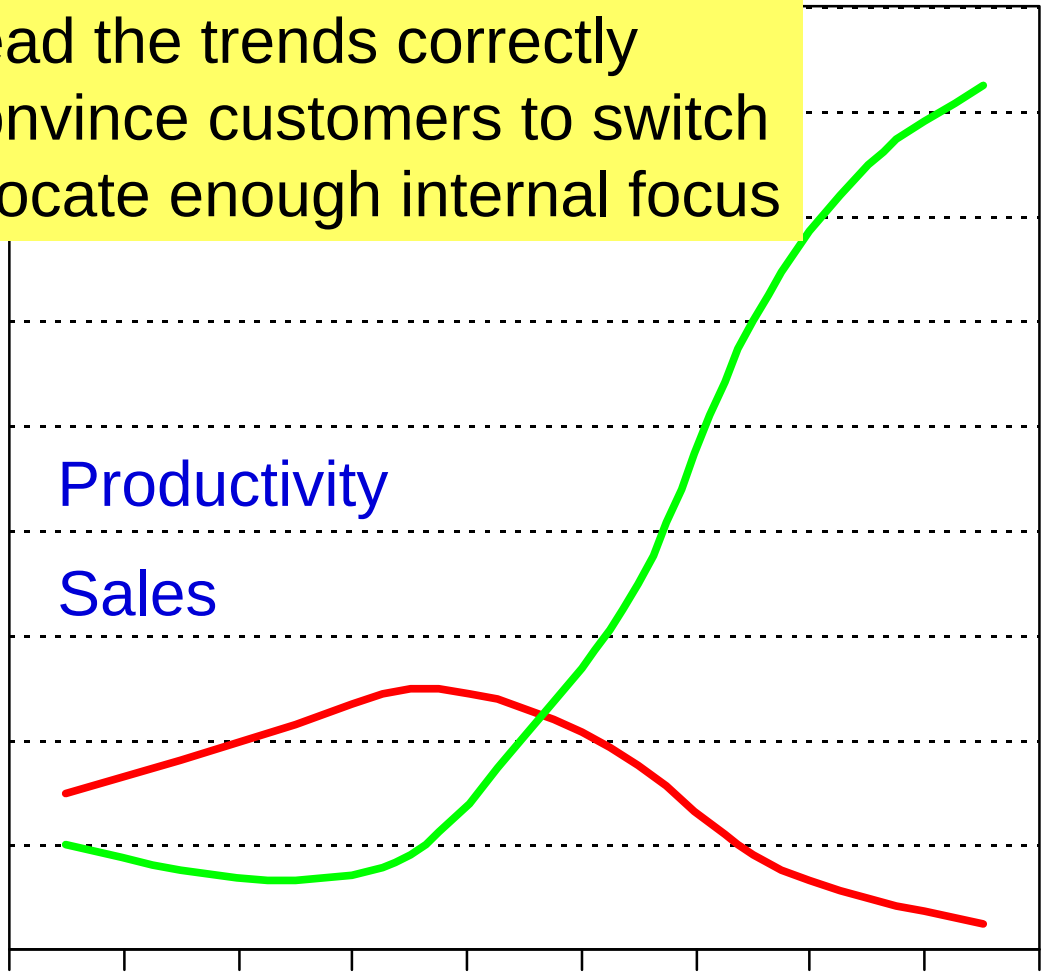
- **"I expect the open-source movement to have essentially won [in] software within three to five years"**
  - **… Eric Raymond, "Cathedral and the Bazaar"**
  - **… 1999**
  - **… (And re-stated in 2001, with the dates clarified as "2003-2005")**
- **"Windows 2000 will not ship in a usable form. (At 60 million lines of code and still bloating, its development is out of control)"**
  - **… "Windows 2000 will be either canceled or dead on arrival. Either way it will turn into a horrendous train wreck, the worst strategic disaster in Microsoft's history."**

Woops!?

# Disruptive trends

It's hard to:
- Read the trends correctly
- Convince customers to switch
- Allocate enough internal focus

*Disruptive technologies are hard to do business with!*




***Inspired by:***
***Clayton Christensen***
**Innovator's Dilemma**

symbian

**Symbian Confidential Internal**

***Market Report: Sea Change in the Tools Industry***

Symbian Product Manager, 12 August 2004

**News from [several companies], and developments around the *opensource* Eclipse IDE platform, indicate a sea change in the tools industry…**

**It is recommended that Symbian respond by wholeheartedly embracing the Eclipse platform…**

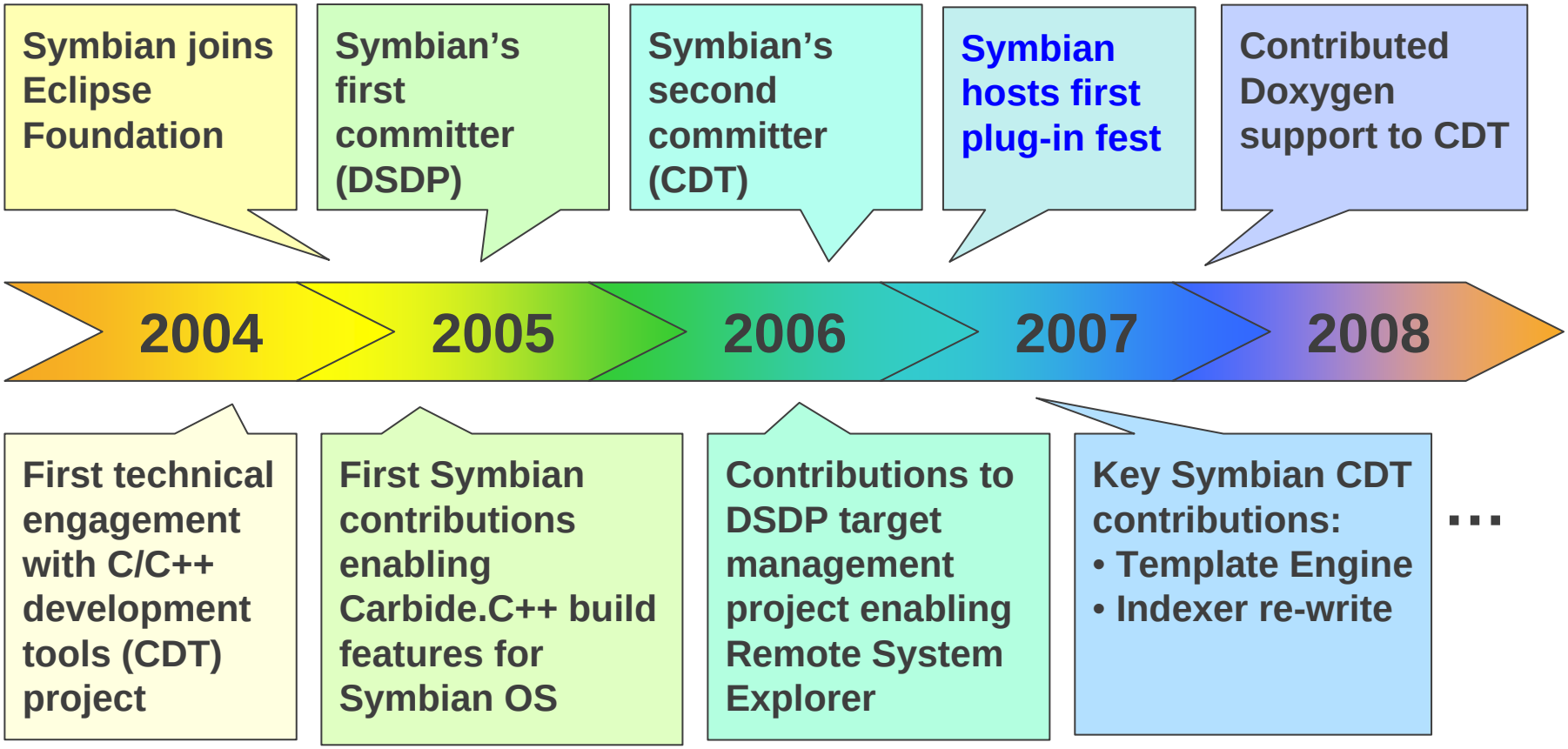
# Some highlights of Eclipse adoption by Symbian

- Carbide family of IDEs for Symbian OS is Eclipse-based
  - ... Launched in 2005
  - ... Used by all Symbian customers & development partners


- Roadmap of improvements
  - ... Latest released version is Carbide v1.3.2 based on Eclipse CDT 4.0
- Symbian is Add-in Provider member of Eclipse Foundation
  - ... Has committers for the DSDP and CDT projects
- Increasing number of developer tools are Eclipse plug-ins
  - ... Used both in-house and in the wider community
  - ... Example: SAW (Symbian Analysis Workbench)


- S60 3.2 phones contain EPL code
  - ... eSWT UI toolkit is part of the MIDP environment on all 3.2 and later S60 devices
  - ... This may be the largest install base for EPL based code

# Some highlights of Eclipse adoption by Symbian



# The first Eclipse Pluginfest

Embedded + Device Development

Sony Ericsson

MOTOROLA

eclipse

NOKIA
Connecting People

LYNUXWORKS™

Telelogic

symbian

WIND RIVER

QNX

IBM

COLLABNET

PERFORCE
SOFTWARE

Mentor Graphics

DIGIA

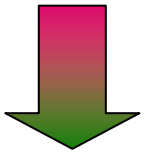
Klocwork

ARM

# Some highlights of Eclipse adoption by Symbian

- Appreciation of the power of a well-governed, meritocratic, open source community
  - … Generates lots of innovation
  - … Support a variety of business models
  - … Competitors co-exist: Community grows
- Gave Symbian world more confidence in the potential to adopt a similar model
  - … Best principles of open source
  - … Licence (EPL) plus governance
  - … Meritocratic yet also business-friendly
  - … Generate lot of innovation
  - … Grow the community

Developer tools system



Mobile device operating system

No panacea – but can be powerful!

# Evaluating the claims of Open Source Software

- Faster time to market
- More eyeballs looking at code, hence higher quality
- Lower barriers to entry, hence greater innovation
- Lower barriers to collaboration
- Lower license fee
- The tide of history…

Difficulties at scale and pace

Fragmentation is easy, Integration is hard

There can be many other costs of development

***And specifically, about Linux:***

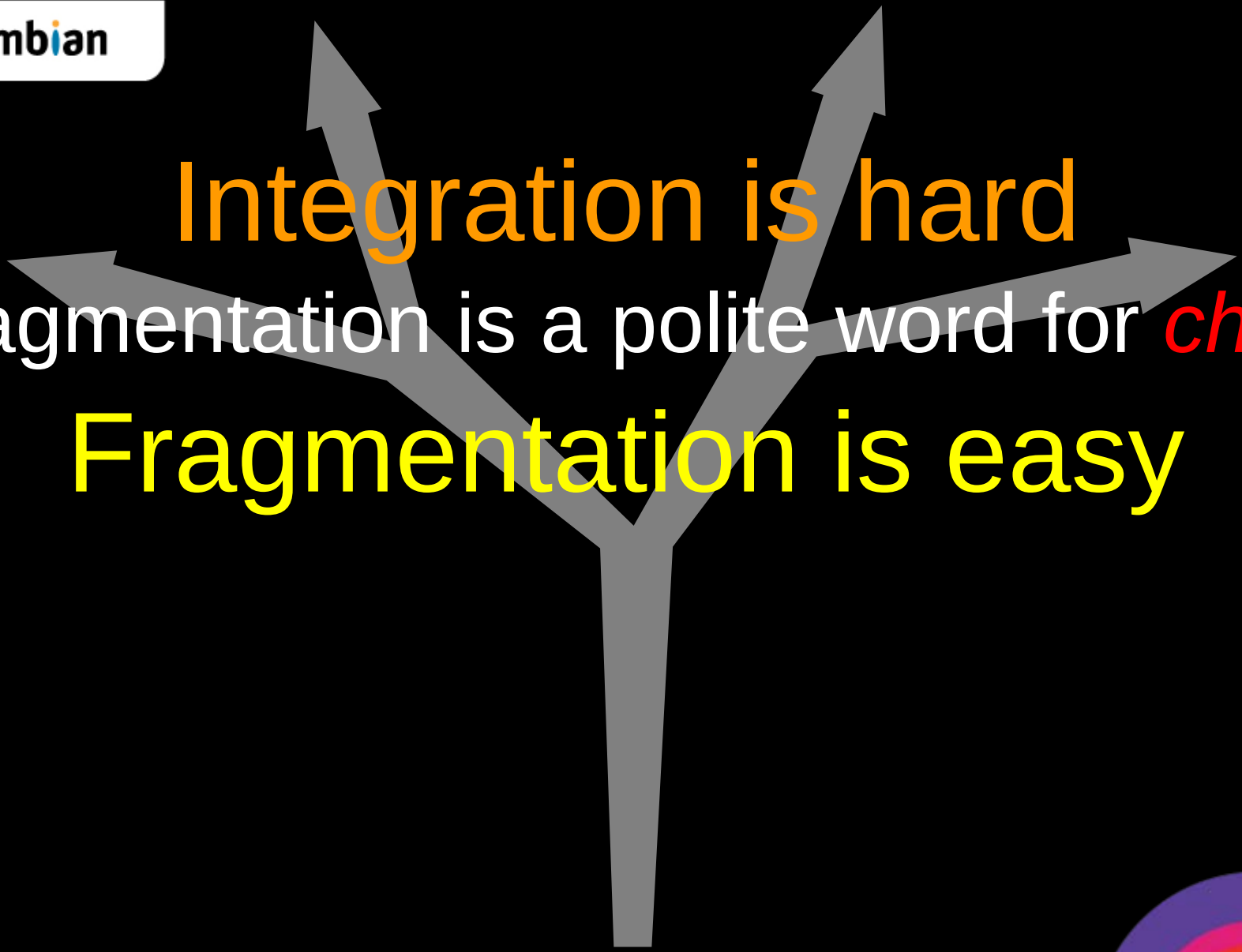
- Better APIs?
- A larger ecosystem – including better tools

Can be copied (eg PIPS/OpenC, Qt)

Not necessarily a better OS

Not conclusive

symbian



# Integration is hard

Fragmentation is a polite word for *chaos*

# Fragmentation is easy

# The six laws of fragmentation

1. **Fragmentation can have very bad consequences**
   - ... Even though there can be good consequences too
2. **Open (or Community) Source makes fragmentation easier**
3. **Copyleft Contracts can help minimise fragmentation**

# The 9 “popular” OSI-approved licenses

- New and Simplified BSD licenses
- MIT license
- Apache License, 2.0
- Mozilla Public License 1.1 (MPL)
- Common Development & Distribution License (CDDL) (Sun)
- Common Public License 1.0 (CPL) (IBM)
- **Eclipse Public License (EPL)**
- GNU Library or “Lesser” General Public License (LGPL)
- GNU General Public License (GPL) (FSF)

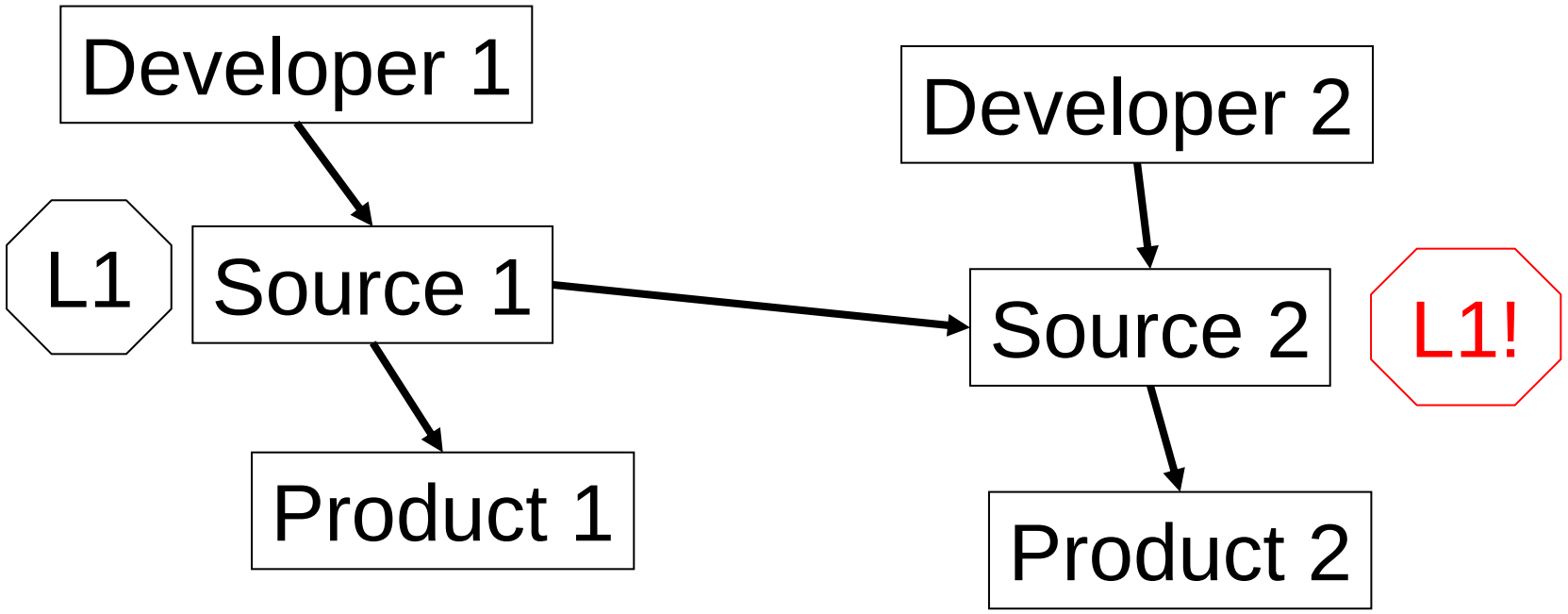
“Academic” / “Permissive”

“Weak copyleft”

“Strong copyleft”

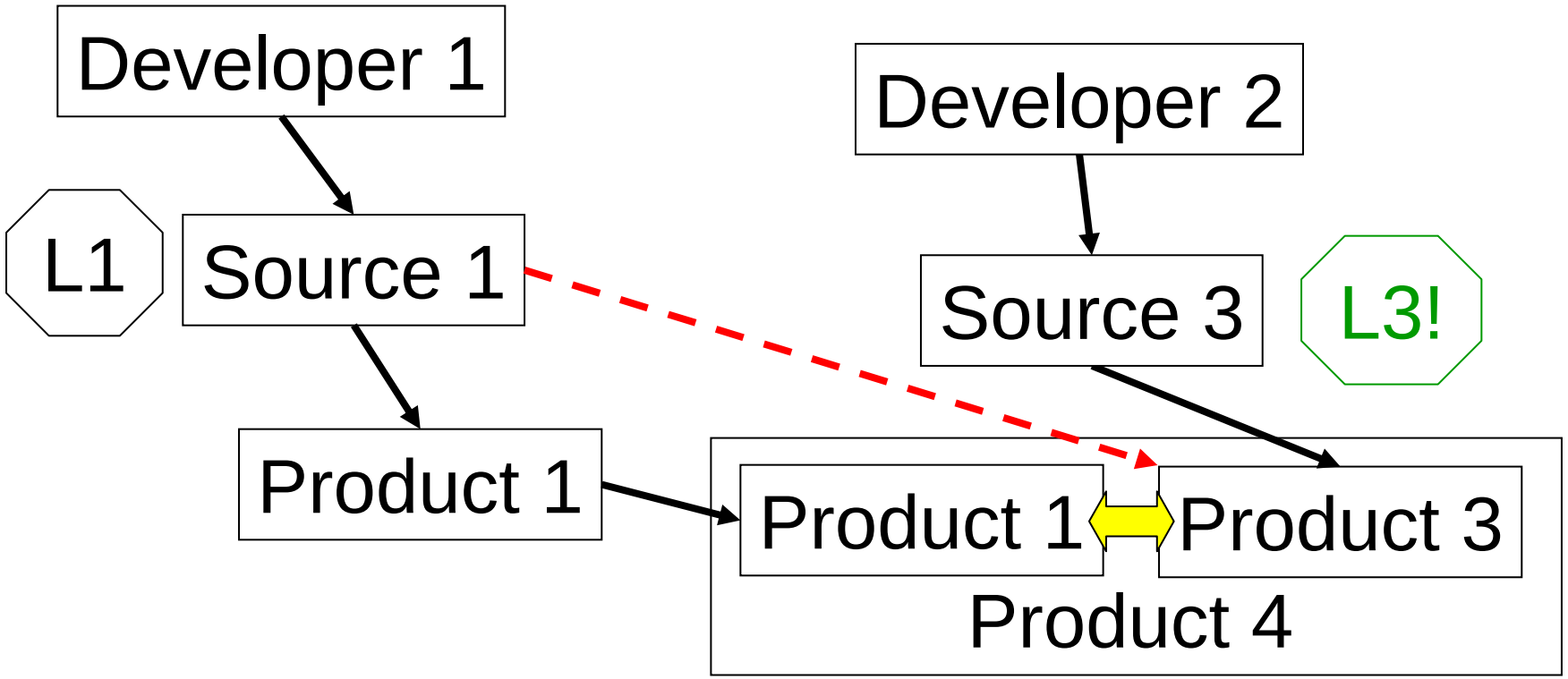
# Weak vs. strong copyleft

- **Modifications and derived works permitted**
- **License must not restrict *other software that is distributed along with the licensed software***



# Weak vs. strong copyleft

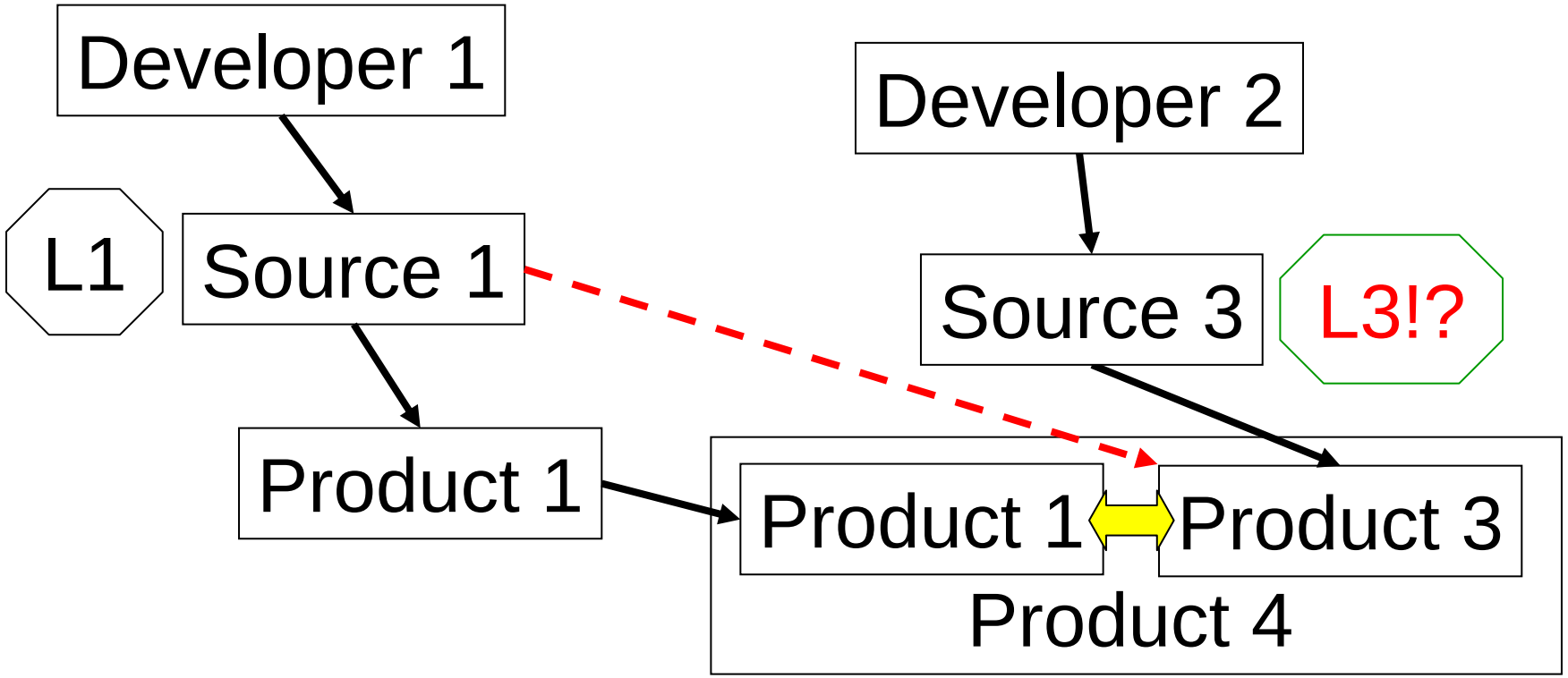
- **Modifications and derived works permitted**
- **License must not restrict *other software that is distributed along with the licensed software***



# Three definitions of weak vs. strong

1. **Focus on “file-level copyleft” (vs. “projects”)**
   - … “Any changes to MPLed files, or new files into which MPLed code has been copied, are Modifications and so fall under the MPL. New files containing only your code are not Modifications, and not covered by the MPL” – MPL FAQ
2. **Focus on “module-level copyleft”**
   - … “For clarity, merely interfacing or interoperating with Eclipse plug-in APIs (without modification) does not make an Eclipse plug-in a derivative work” – EPL FAQ
3. **“For library use” (LGPL) vs. “For general use” (GPL)**
   - … LGPL: Software that links to the library can have a different license
   - … GPL: Software that links to this component must also use GPL
   - … **FSF say they prefer people to use the GPL**

# Motivations for and against strong copyleft

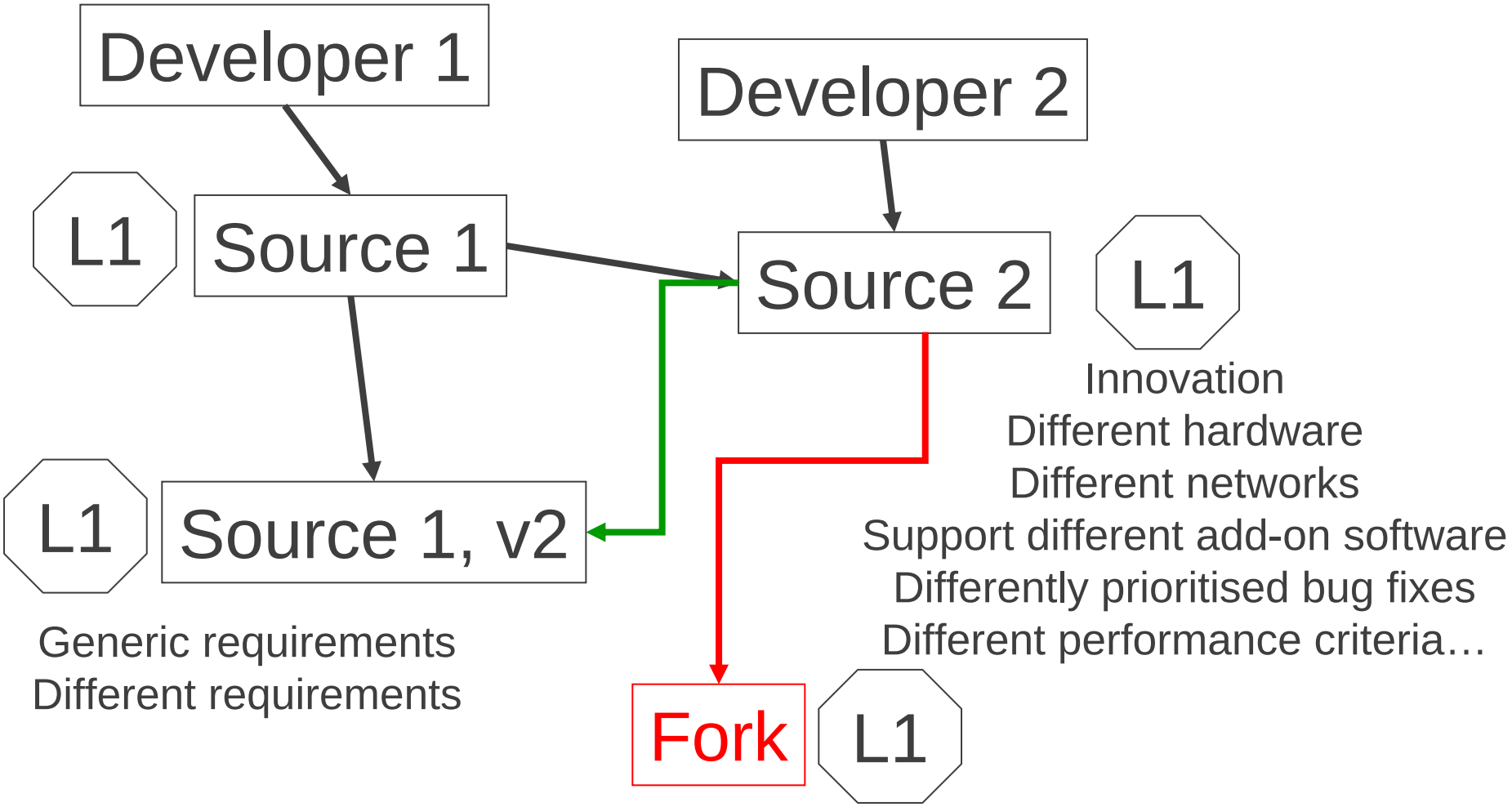



- Developer 2 wants to be able to make money from royalty fees – or preserve some trade secret in Source 3
- Developer 1 wants to encourage innovation (from Developer 2)
- Developer 1 may want to avoid "**hoarding**" and "**fragmentation**"

# The six laws of fragmentation

1. **Fragmentation can have very bad consequences**
   - ... Even though there can be good consequences too
2. **Open (or Community) Source makes fragmentation easier**
3. **Copyleft Contracts can help minimise fragmentation**
4. **Fragmentation can't be avoided simply by picking the right contract (and enforcing copyleft)**
   - ... The integrators may reject changes made by individual developers

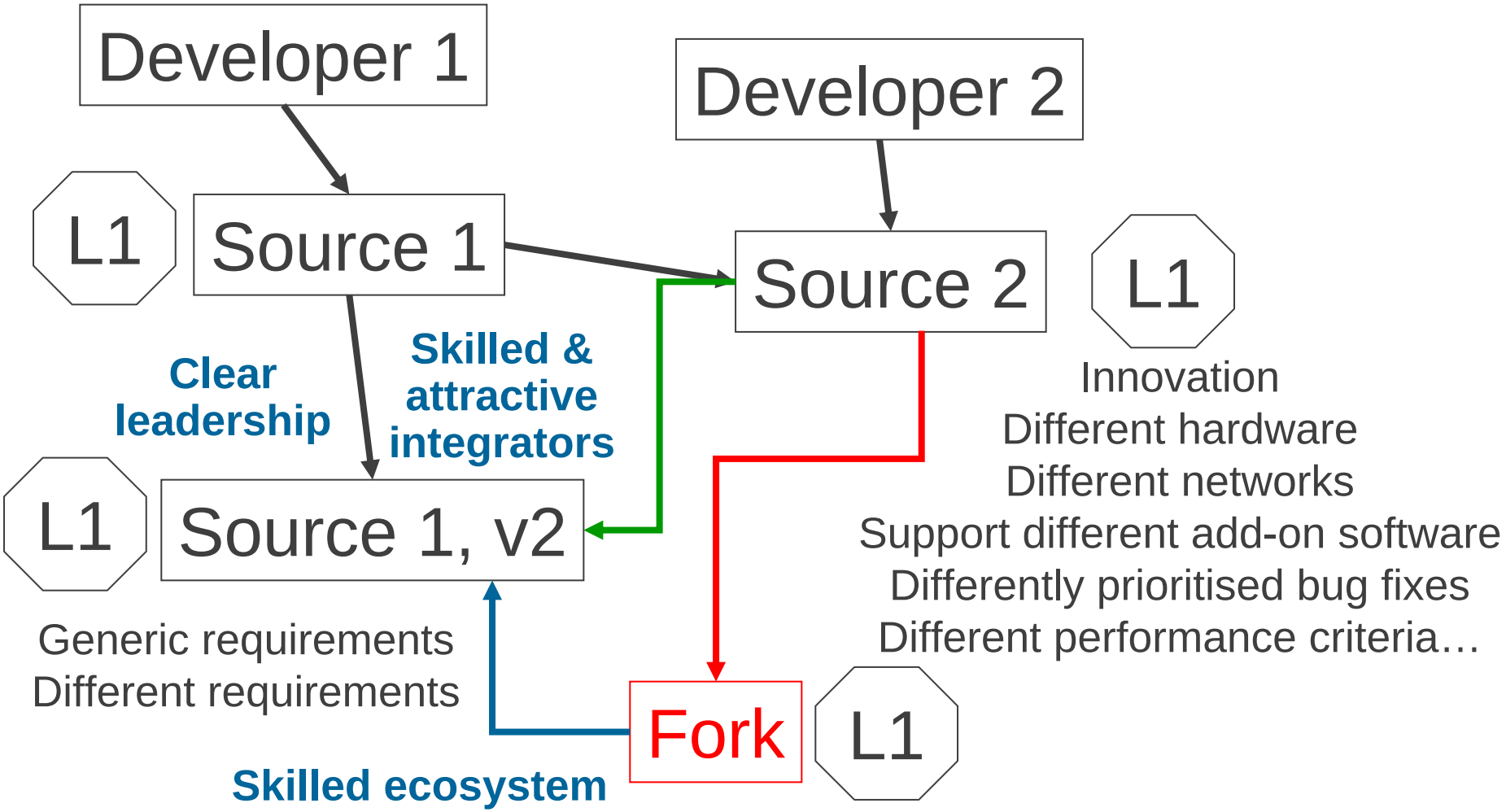
# Fragmentation even with copyleft



# The six laws of fragmentation

1. **Fragmentation can have very bad consequences**
   - … Even though there can be good consequences too
2. **Open (or Community) Source makes fragmentation easier**
3. **Copyleft Contracts can help minimise fragmentation**
4. **Fragmentation can't be avoided simply by picking the right contract (and enforcing copyleft)**
   - … The integrators may reject changes made by individual developers
5. **The best guarantee against platform fragmentation is powerful platform leadership & expert ecosystem**
   - … Trustworthy: well-motivated and competent

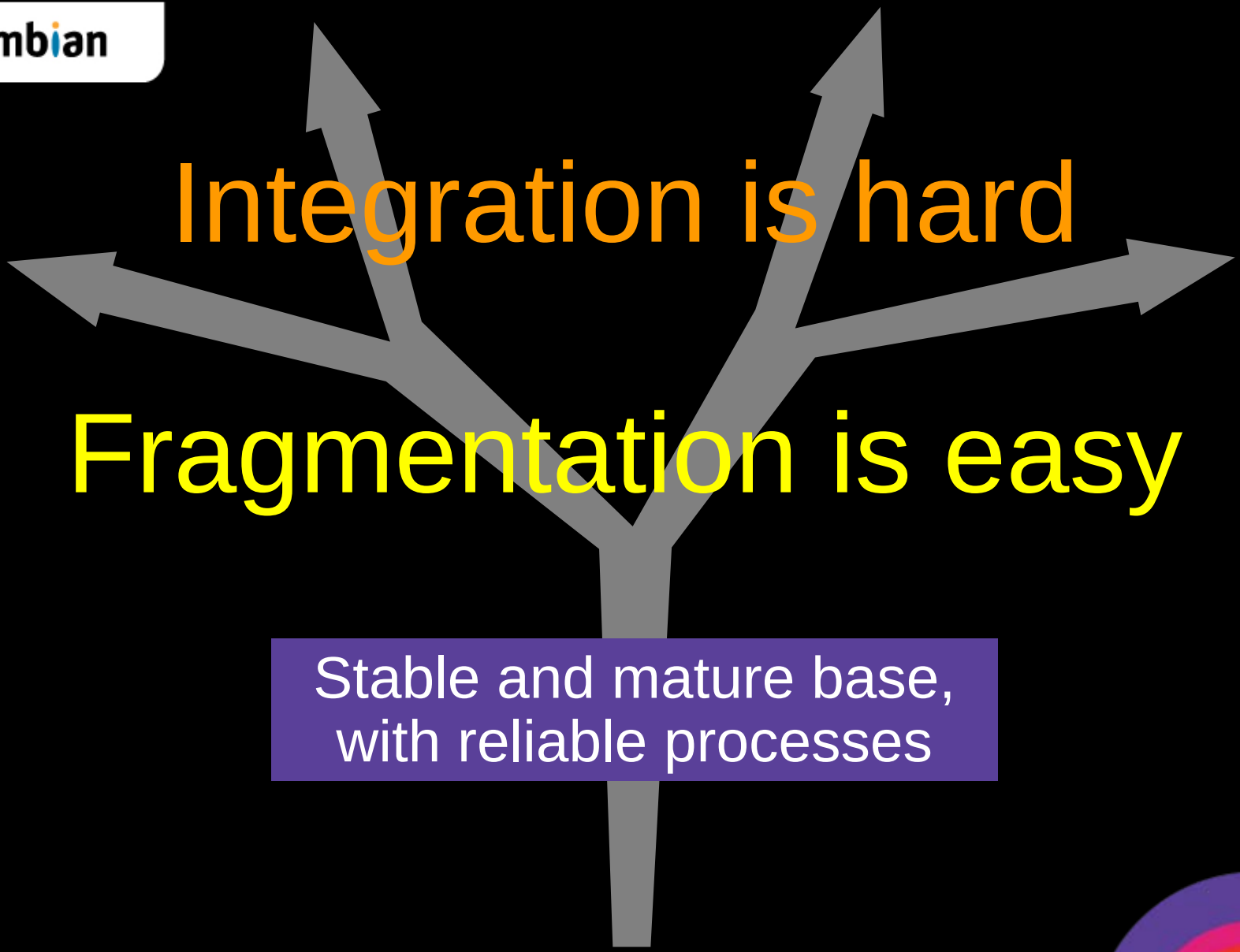
# Overcoming fragmentation tendencies



# The six laws of fragmentation

1. **Fragmentation can have very bad consequences**
   - … Even though there can be good consequences too
2. **Open (or Community) Source makes fragmentation easier**
3. **Copyleft Contracts can help minimise fragmentation**
4. **Fragmentation can't be avoided simply by picking the right contract (and enforcing copyleft)**
   - … The integrators may reject changes made by individual developers
5. **The best guarantee against platform fragmentation is powerful platform leadership & healthy ecosystem**
   - … Trustworthy: well-motivated and competent
6. **The less mature the platform, the more likely it will be to fragment, especially if there's a diverse customer base and a hectic market environment**

symbian



# Integration is hard

# Fragmentation is easy

Stable and mature base, with reliable processes

***Which mobile OS best solves the series of hard problems for device & service creation?***

And delighting them

Large, viable market

Best principles of open source software

Nimble and **agile** response to market needs

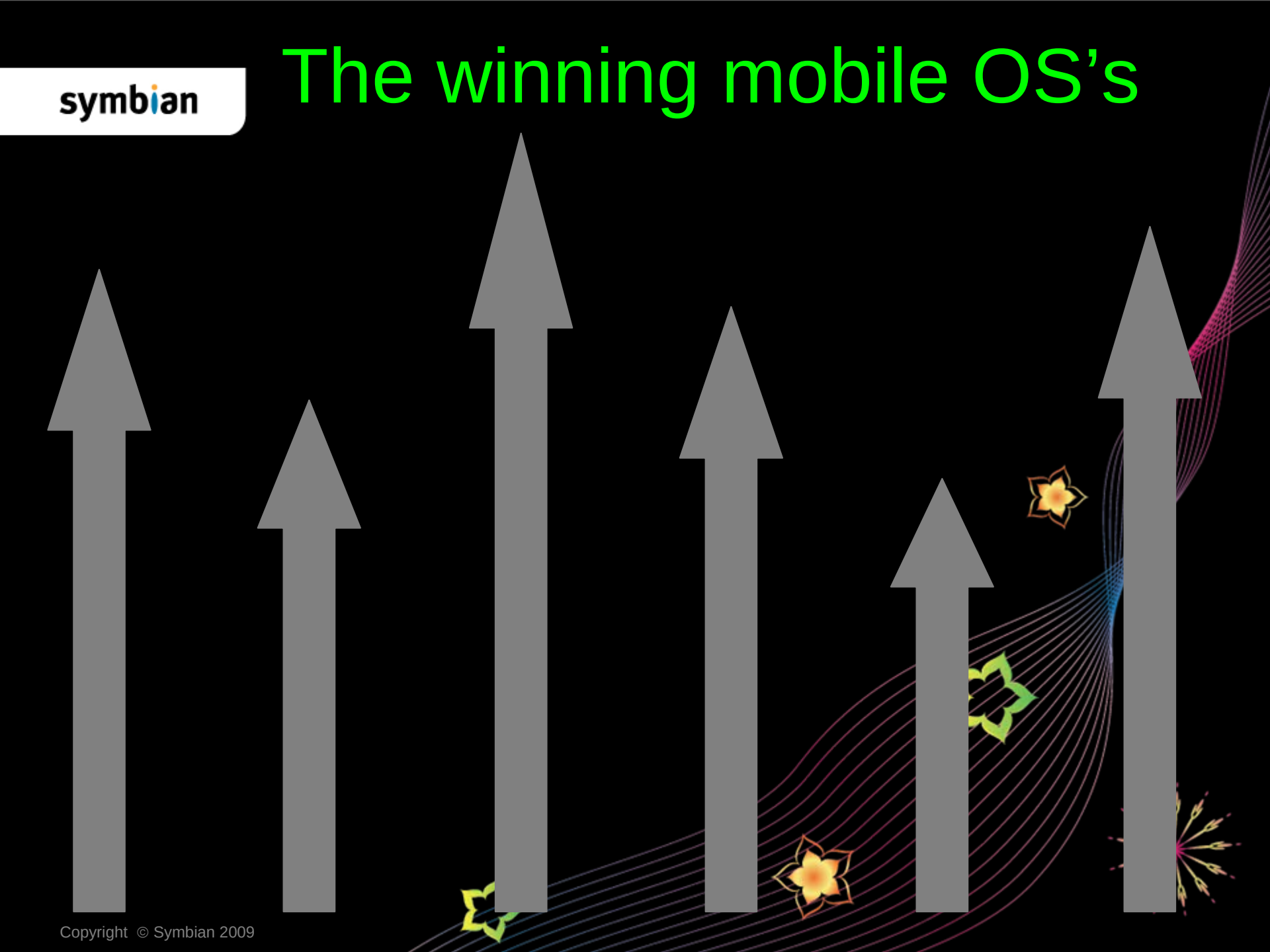
Large pool of productive and engaged **developers**

Intensely listening to customers

**Stable** and mature base, with reliable processes

Tools, APIs

# The winning mobile OS's



OS diversity can be hidden from applications by intermediate layers (eg Java)

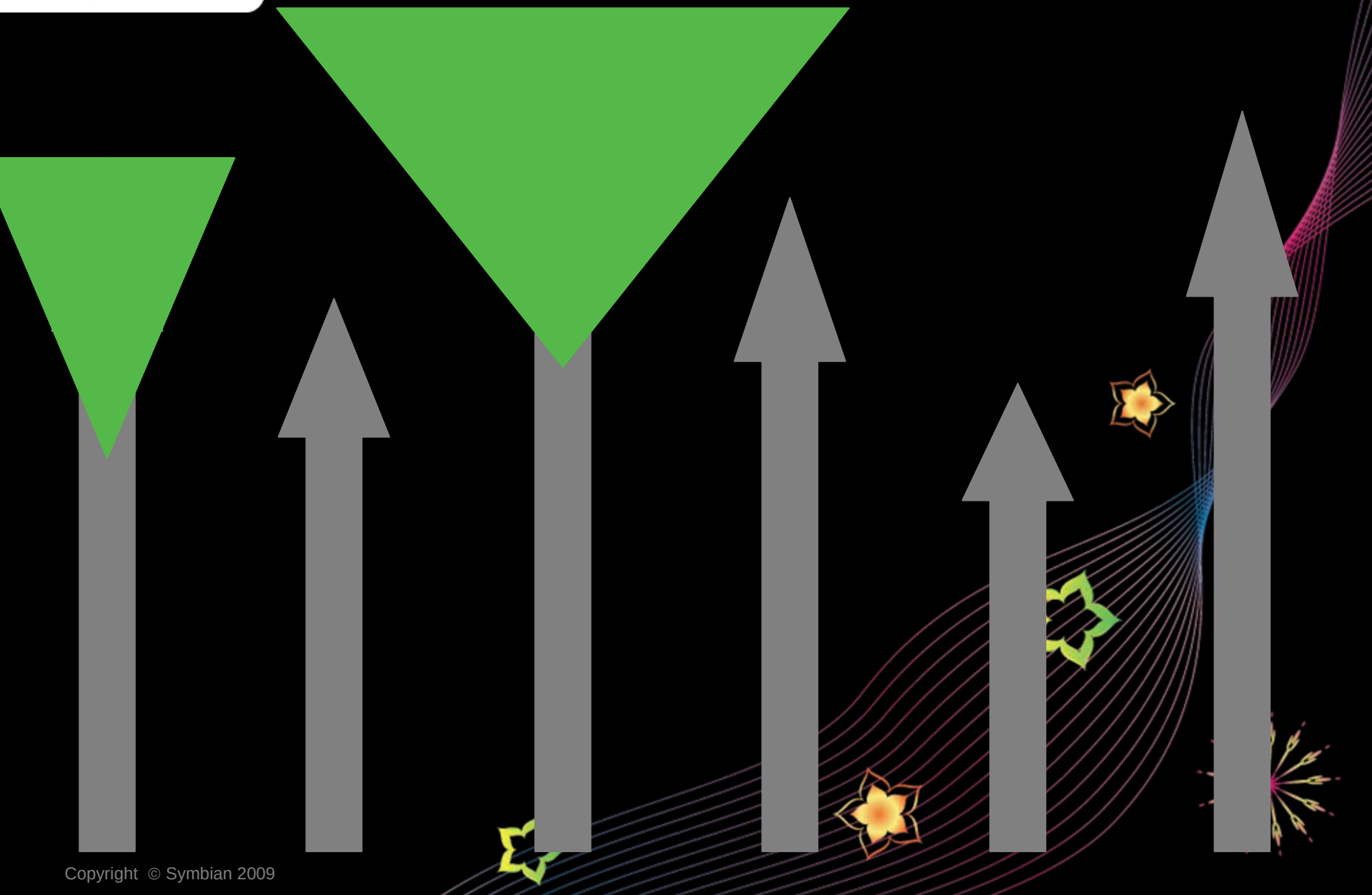
Some apps do need native access (for speed, APIs…)

?

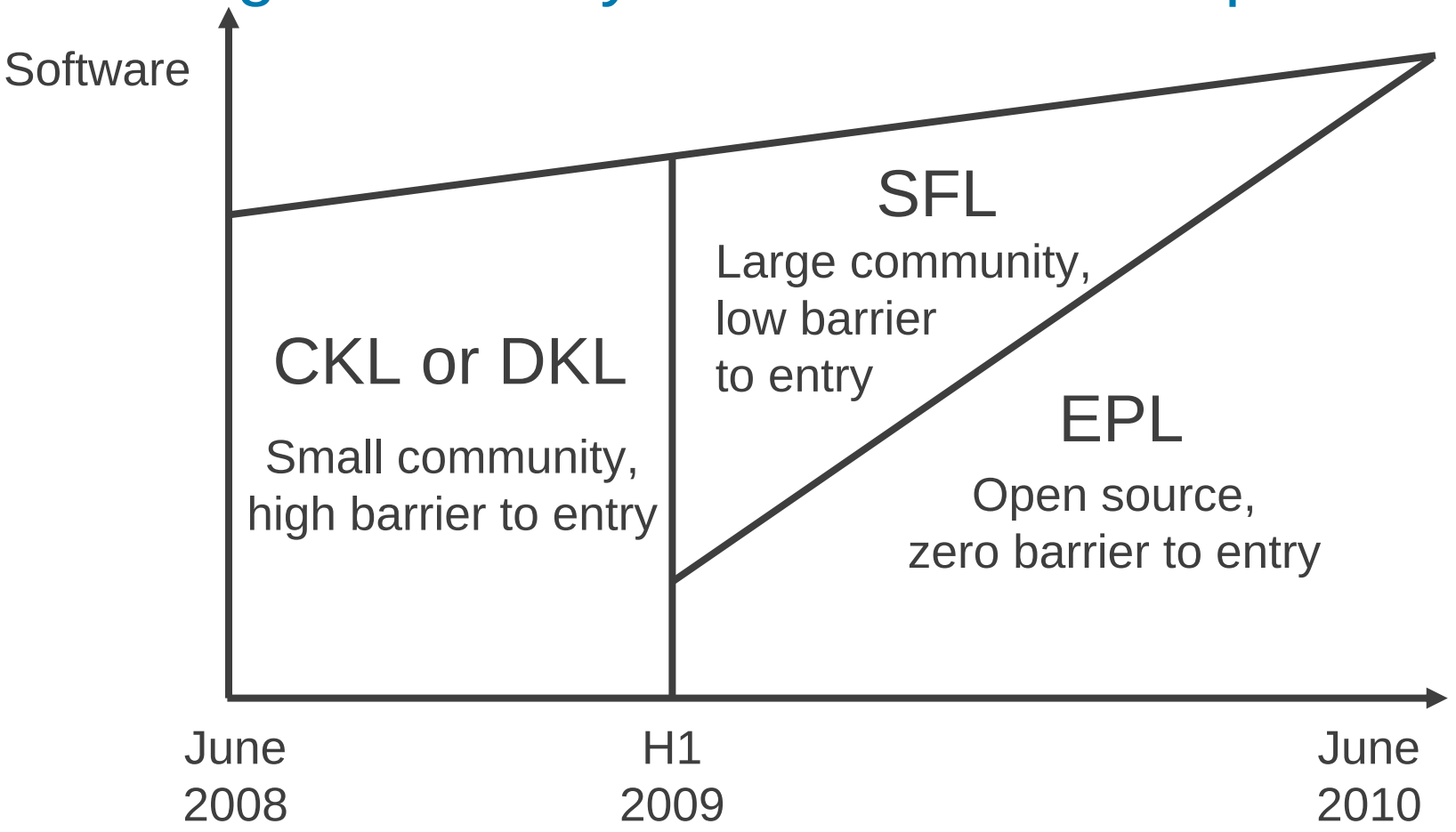
The *device creation community* needs consistent and reliable native access

Device creation fails if the DCC experiences too much OS fragmentation

# The winning mobile OS's



# Proving time for Symbian Foundation platform

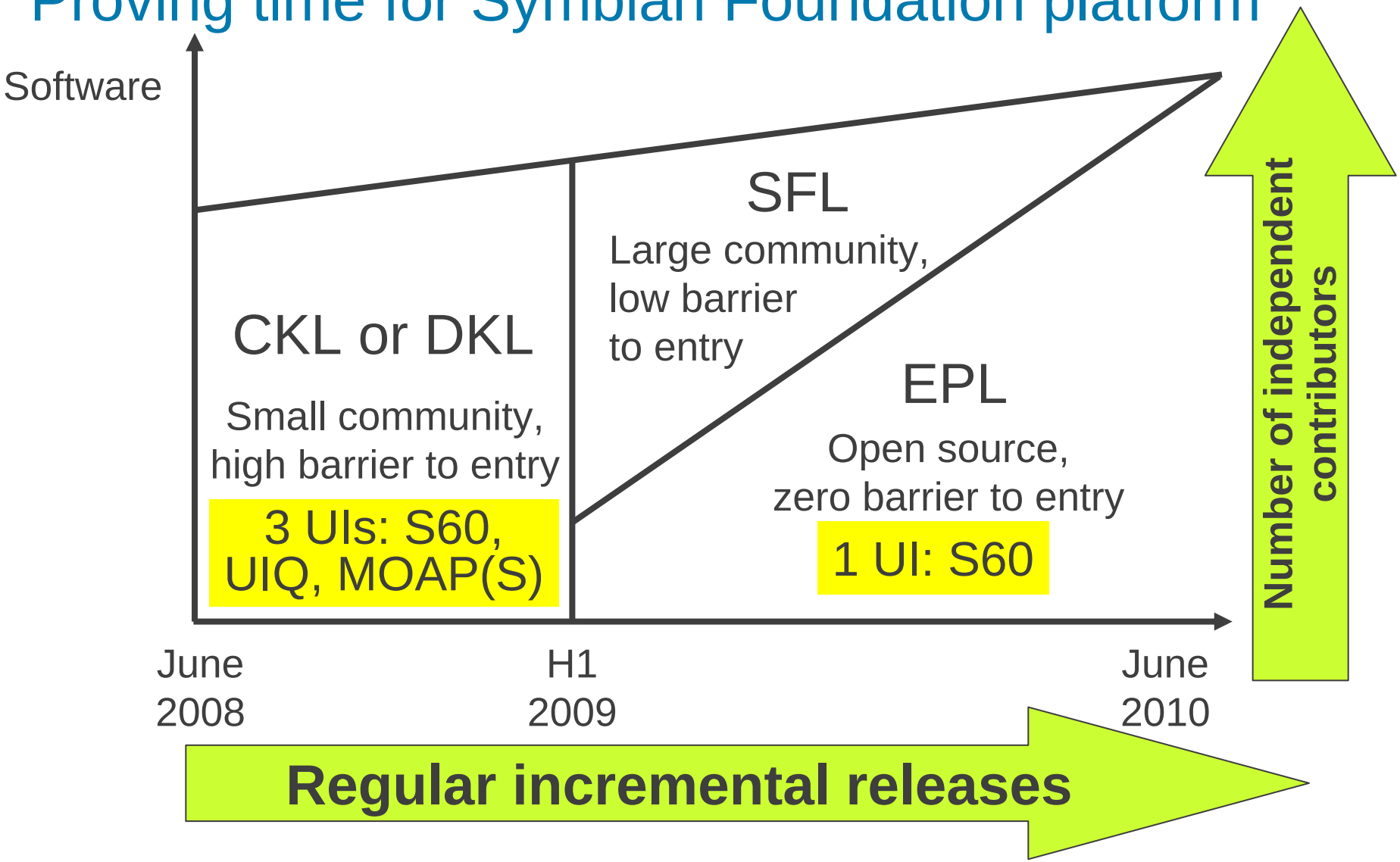


# Why the EPL? – choosing an OSS license

- The choice of a license is a necessary (but not sufficient) part of reducing software fragmentation
  - … This is an argument for a “weak copyleft” license
  - … Permits the creation of new components – above, below, or alongside
  - … Stops customers from hoarding their changes to original components

1. The license should be “**business friendly**”
   - … Clearly written; Already tried and tested
   - … Minimises the risk of customers being unexpectedly forced to release source code for their own innovative new components
2. The license should be “**integration friendly**”
   - … Supports coexisting with software written under other licenses
3. The license should be “**patent friendly**”

# Proving time for Symbian Foundation platform



# Symbian Foundation supporters

# Symbian Foundation governance principles

- Cooperate on the shared infrastructure
  - … Enable innovation and competition outside shared infrastructure
- Contribution is king
  - … Expect deeper contributions from wider and wider groups of people
- Meritocracy rather than $$$ decides

**Transparency of process as well as of code**

# Expand huge business opportunity

## Symbian Foundation operating principles

- Role of Foundation itself is to create huge leverage
  - … Limited software development by Foundation staff members
    - Software codeline management
    - Developer ecosystem management
- Councils to elevate best ecosystem understanding
  - … Roadmap planning; Architecture;
  - … UI and usability; Releases
- Regional Advisory Councils
- Keep the platform the clear leader
  - … Minimise the risks of platform fragmentation

# Four top Symbian Foundation priorities

1. Quickly complete highly competitive fully open mobile operating system
   - … Mobile software set free…
2. Enable easy creation of rich applications
   - … Web, Python, Java, Flash, Silverlight, Qt Software, native…
3. Enable easy introduction of novel hardware
   - … New kinds of smart mobile device
   - … New peripherals and new hardware providers…
4. Remove friction from the wider ecosystem
   - … Streamline routes from innovation to market
   - … Improve application signing, application deployment…

***Accelerate consumer experiences***

symbian

# Success metrics 2008-2013-2018