OSS in the industry : the THALES example

Jean-Michel Tanneau

August 2004

jean-michel.tanneau@thalesgroup.com

Corporate Department

THALES

The point of view : THALES

- ⇨ (Software dominant) Systems integrator

The context

- ■ Increase of complexity & Price reduction
- ■ Conflicting lifecycle : Technology - COTS Versus Systems
- ■ Strong requirements : Reliable, Secure, Flexible, Configurable, Scalable, Available & Maintenable in LT
- ■ Small volumes
- ■ COTS era (Perry directive)

Objectives

- ■ Increase performance (**effectiveness**): *quicker, better, cheaper*
- ■ Improve durability of R&D investments (core business)

R&D software strategy : 2 of the priorities

- ¬Open architectures & Standardization
- ¬Sharing & cooperation on generic technologies (non core business)

Is OSS an opportunity to meet objectives and R&D strategy?

- How to benefit from the product ?
- How to benefit from the development process ?
- How to benefit from the mechanisms of « value creation » ?

OSS & Thales

Two phases

- Since 1999 : Usage of OSS (in business)
  - Main focus : To control risks
- Since 2002 : Use of OSS as a process
  - Main focus : To leverage opportunity

One approach, a mix of

- Strategic approach and (Technical) Change management

How to benefit from

OSS as products (technical objects)

Why to take an interest in OSS? COTS drawbacks

- Uncertainty
  - Product (black box) and delivered information (claimed Vs actual behaviour)
  - (product and editor) Strategy : evolution, roadmap, business model
  - Market (continuous restructuring)

- Subordination to a sole provider (monopoly)

- Divergent interests
  - “mass market” driven : progressive disinvolvement with our business
    - Certain domains considered as « niche » market
  - Shortening of COTS life cycle – Impact on Quality

- Others
  - COTS is intrusive (architecture / design)
  - Support
  - Cost ?

Use of OSS - Stakes

Opportunity

- Same advantages as COTS : productivity (time to market-cost to market) & added-value
- New source of provisioning (opens the market)
- Providers independence (durability of components) - **Control over system life cycle**
- A spreading (free & competitive) supply
- Trends : support from large IT companies & institutional users (administrations & MoD)
- Community based AND commercial **support** (free & competitive market)
- White box : secure (auditable), adaptable, predictable (certification)
- White process : evolution, quality (fast bug corrections)
- Users & technology driven
- **Standards based (Interoperability) - Commodification**
- !!! TCO ?

Threats / Brakes

- IPR (!!! **OSS licenses**)
- Warranty and Liability
- Software patents - LZW (GIF), MP3, SCO Vs. IBM lawsuit
- Diffuse and **unequal (quality) supply** - (!!! care to not generalize: OSS is not a "guarantee of quality")
- Still an **external component**
- **Continuous evolution**
- Mixing many OSS
- Complexity (skills/training required)
- Un-grasped world + FUD

Legal issues

License analysis

- Is it an OSS ? (OSD compliant)
- Existence of third party patents ?
- Identification of restrictions / conditions related to redistribution (with or without modification)
- If clarification needed, apply to the author

Usage in a specific programme

- Usage is analyzed & documented (software architecture)
- Compliance with contractual requirement and regulation
- If many OSS used, check that their licenses are compatible

The Component Evaluation / Selection process

Aims

- reduce and/or delay risk occurrence - mitigate impact
- Effectiveness : fulfills technical requirements
- Confidence (now & mid-term)
- Economic efficiency : TCO, know-how capitalization (ROI)

⇨ To get *“The right product, at the right time, at the right cost and available for the right period”*

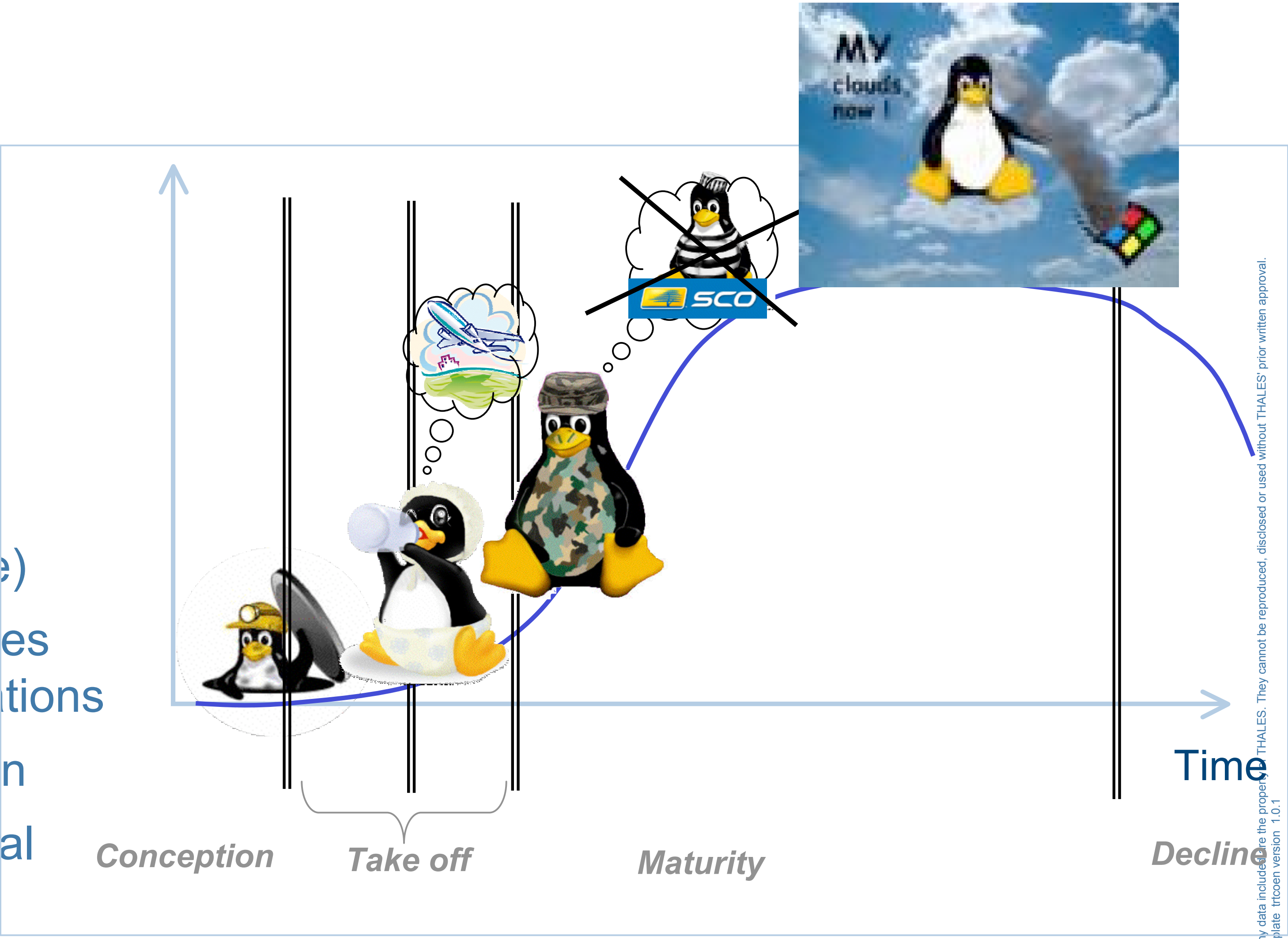
Approach

- Technical assessment
- Industrial assessment

OSS dedicated Evaluation process

Investigation

- Project (life)
- Communities and motivations
- Competition
- Professional support market



OSS dedicated Evaluation process

Emergence

- Initial objective
- Initiators
- Origin of the project
- Positioning
- 1st stable release

Taking-off

- Major events
- Successive stable releases

Maturity

Understanding

- Current model of development
- Developers community
- Positioning
- Adherence to standards
- Licensing scheme

Support

- Mailing lists
- FAQ
- Forums
- Commercial companies ? costs ?

Acceptance

- Echo in the press
- Related web sites
- Industrials supporting the development
- Number of users
- Institutional & industrials users

Future

- Roadmaps

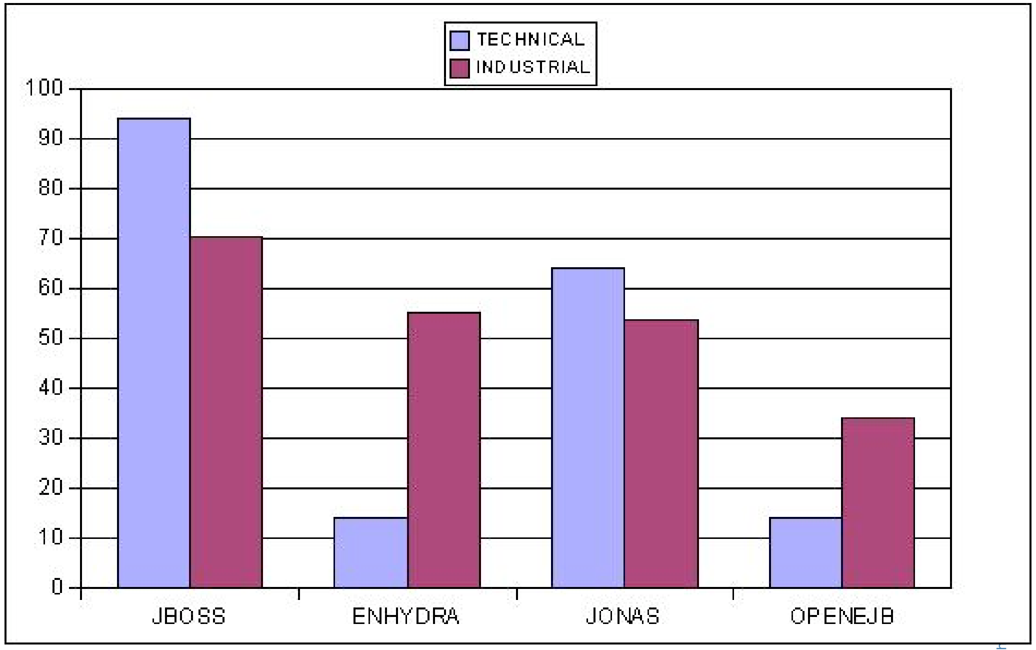
Evaluation - J2EE sample

Technical (aggregated) criterion & Weight

EJB 2.0 support	**50**
CMP 2.0 support	**30**
Database support	**20**
Total	*100*

Industrial (aggregated) criterion & Weight

Professional technical support	**20**
Users population (nb, role)	**15**
Project (re-)activity (Q/A Mailing lists)	**13**
Release & correction frequency	**12**
Company hosting	**10**
Developers community	**8**
Information (Web & Forums)	**7**
Documentation	**6**
Relationships with other OSS	**5**
Press/web footprint	**4**
Total	*100*



Process run & assessed
on 2002 (MILOS project)
- 350 OSS (45 segments)
- Results published on **eCOTS portal**

Part 1 : Conclusion

Introducing OSS in the scope is a change, then, organizational changes are needed

- A corporate policy to control correct usage of OSS
- A dedicated team (multi units)
  - to provide legal analysis, advice and audits
  - to capitalize and to organize technical exchanges (workshops, lessons learnt) ; to set up networks of experts (evaluation process)
  - to make known issues (awareness campaigns to all stakeholders)
  - to ensure the smooth running of local organization (enterprise level) in charge of procurement, validation/qualification, deployment, configuration management
  - to survey external expertise (support market)
- Updates of corporate referential
  - Components evaluation & selection guideline
  - Components usage guideline

How to benefit from OSS:

- through the collaborative development model
- through the process

Benefit from the OSS development model

A model to improve quality, productivity & collective intelligence

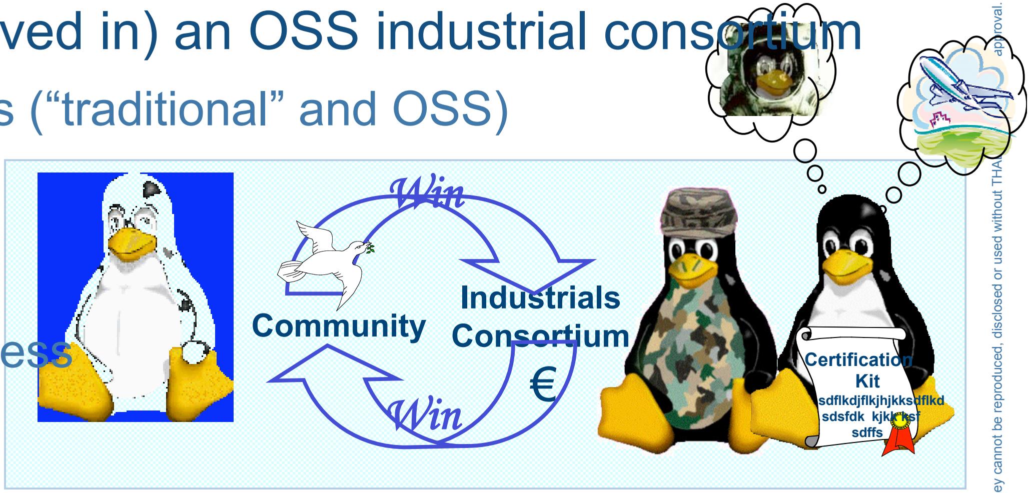
- Thales Collaborative Development Platform
  - Experimentation started mid 2002
  - 25 projects - 60 active developers
- Experimentation assessment on-going, preliminary results show benefits related to:
  - Reuse (& convergence)
  - Synergy - Sharing & co-operation
  - Quality (peer reviews)
  - Personal motivations (recognition)
  - Technological communities
  - Unified project referential for all artefacts (source, doc, mail, news, bug tracking…)

A great disruption : from local (department/division/unit) to corporate interest

Benefit from the OSS process

A mean towards open architectures and standardization
perpetuate R&D investment

- Launching of (or getting involved in) an OSS industrial consortium
  - take the best of the 2 worlds (“traditional” and OSS)



- Key success factors
  - Need covered & Attractiveness
  - Motivation
  - Licensing schema
  - Budget - Plan
  - Team building & Project management
  - Consortium building : Partnerships and strategic objectives
  - Community management

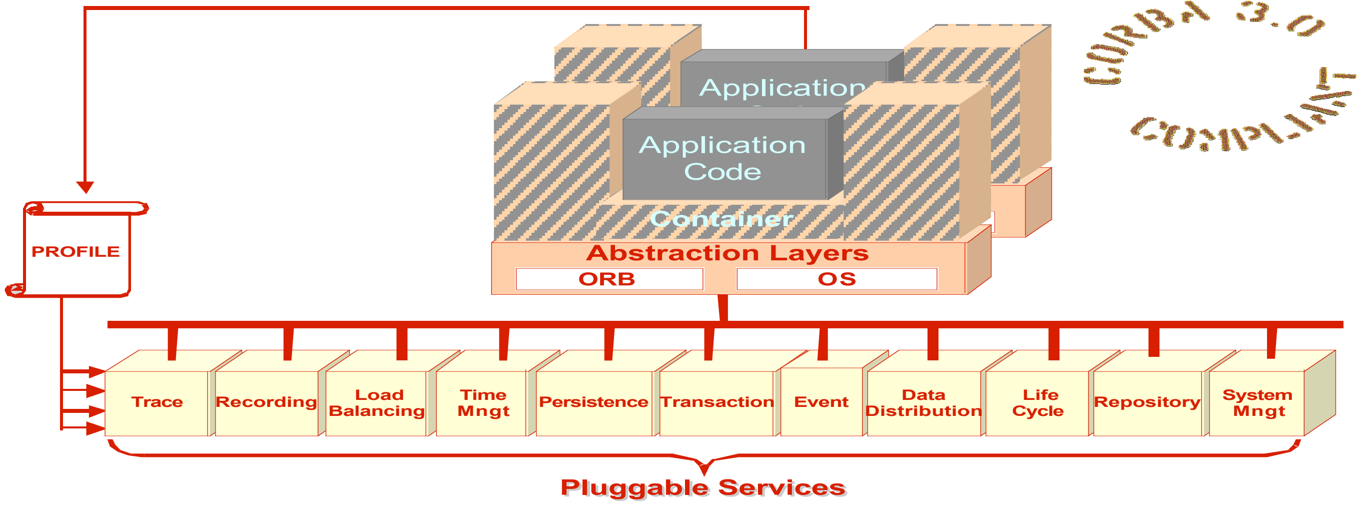
Make IP free (OSS) is not as easy as it could seem

CARDAMOM

A FRAMEWORK FOR NEAR REAL-TIME & FAULT TOLERAN
COMMAND CONTROL AND INFORMATION SYSTEMS (

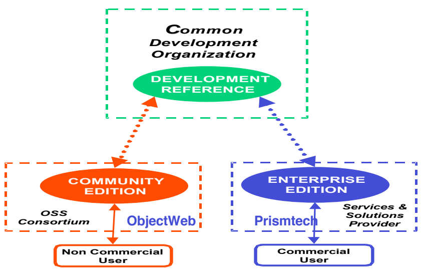
under **Open Source** distribution

AN OPEN AND FLEXIBLE PLATFORM



CORBA 3.0 COMPLIANT

DISTRIBUTION SCHEME



A COMMON DEVELOPMENT ORGANISATION

Creation of an **Industrial Organisation** currently grouping

THALES EMA

Open Structure ready to welcome new partners

FOUNDATIONS

Civil

Secure Op.

Defense

1 MULTI-DOMAIN

2 CORBA Services (e.g. Data Distribution Service)
COTS integration or specific value-added implementations that fit industrial needs

3 Support of CORBA Component Model extended to CCIS requirements

4 Active participation to standardisation of middleware sources for CCIS

5 Support of Model Driven Architecture through the use of a UML tool chain

MULTI ENVIRONMENT

REFERENCE PLATFORM

LINUX redhat LINUX C++ TAO

LINUX redhat LINUX JAVA COMPATIBLE JacORB

MULTIPLE POSSIBLE COMBINATIONS

Target Platforms:
- Linux RedHat
- SUN Solaris
- AIX

Languages:
- C++
- Ada 95
- Java

Object Request Brokers:
- TAO
- JacORB
- OrbRiver Ada

LINUX redhat LINUX Sun microsystems SOLARIS IBM AIX JAVA COMPATIBLE ADA 95 C++ TAO JacORB OrbRiver

JMT / WCC Toulouse 2004

Conclusion 1

OSS

- is there,
- is not a marginal phenomena
- comes with opportunity

- is disruptive :
  - changes in organizational structures : formal ones (units in charge to analyze risks, to recommend, to deliver, to maintain) and informal ones (networks of experts)
  - changes in organizational techniques (business referential) : purchase, business management, design, development integration, test/validation, deployment, maintenance

Use (**correctly**) when it makes sense
COTS & OSS have a place in systems

Conclusion 2

OSS allows :

- Improvement of performance
- Improvement of R&D investments
- Standardization (commodification) of software architecture
  (openness, interoperability, technology insertion)
- Sharing & cooperation on non-core business technology

OSS is **A** mean to meet our R&D strategy