The use of Free Software (Freeware) in Thomson-CSF

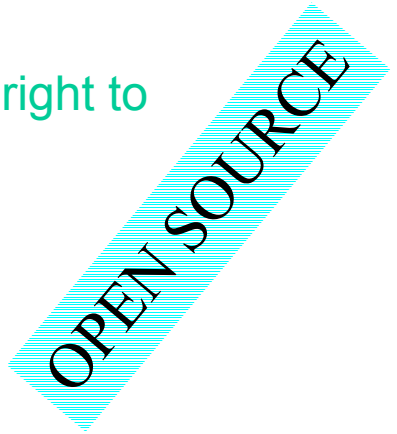
Jean Aguado
CET Software
Thomson Technologies & Methods

Objectives

- ➔ Benefit from open source
- ➔ Eliminate potential risks
- ➔ Increase awareness

What is usually called freeware ?

- **Free as "free lunch" (black boxes)**
  - **Binaries that can be used without having to pay :**
    - Examples :
      - Internet Explorer
      - Acrobat Reader
- **Free as "freedom" (white boxes)**
  - **Programs provided with their sources** and a license that grants the right to **inspect, modify** and **distribute** these sources.
    - Examples
      - Linux Operating System
      - Apache Web Server
- **Not really "free" : Commercial products**
  - **Free evaluation programs**
    - After a given evaluation time a fee is due for its use
  - **Programs free of charge for personal use but requiring the payment of a fee in professional use.**
  - **Shareware**



September 1999
Creation of the Freeware Working Group

- **Participating:**
  - **TPI** (M.O. Warrik; G. Lefranc),
  - **Airsys, Airsys ATM, NCS, TCC, Detexis, Syseca, TCO**
- **Identified advantages of open source software**
  - Ability to inspect , understand , fix the problems in the sources
  - Long term support
  - Higher quality than the commercial products
  - Widely available on the Web
  - Portable environment
- **About 120 different freeware in use in these 7 Units**
- **60 to 70 % of these freeware use the GPL (General Public Licence) of the FSF (Free Software Foundation)**

Why freeware works

➔ Individuals : Recognition

- Industrial software development is often anonymous. Some developers (among the best) feel frustrated about this and want to be recognised by their peers.
- Some developers feel that access to the source is just like access to the knowledge. It has to remain free. Keeping the sources secret kills the evolution of the software.

➔ Enterprises : New economy model based on Services

- When the important part of the business is the service, traditional companies don't have major concerns about providing services around freeware.
  - Ex: IBM, HP, SGI, NETSCAPE, etc

Selling models for large distribution software

Traditional (Microsoft) The software market is a **product market**	**Open Source (Netscape)** The software market is a **service market**

• The service is just there to support the selling of software
  • *Minimal services*
  • *Light infrastructure (distributors)*
  • *Permanent innovation*
  • *Restrictive licensing policy*
  ◆ *tokens*
  ◆ *non transferable*
  ◆ *temporary ?*
	• The software is given for free because it creates the need for services.
	  • *Minimal distribution cost (Web)*
	• The notoriety of the software and/or its creators is the real source of revenues
	  ◆ *fame; brand image*
	  ◆ *consulting*
	  ◆ *conferences*
	  ◆ *services*
	  ◆ *employment*

• Advertise the features • Protect the know how • Implement mechanisms in order to restrict uncontrolled diffusion (copy, piracy)	• Demonstrate the know how and skills (Sources, FAQ's, on line discussions) • Protect your public image • Implement incentive mechanisms to ensure a wide diffusion of the software (license, web)

What is today the selling model for Thomson-CSF ?

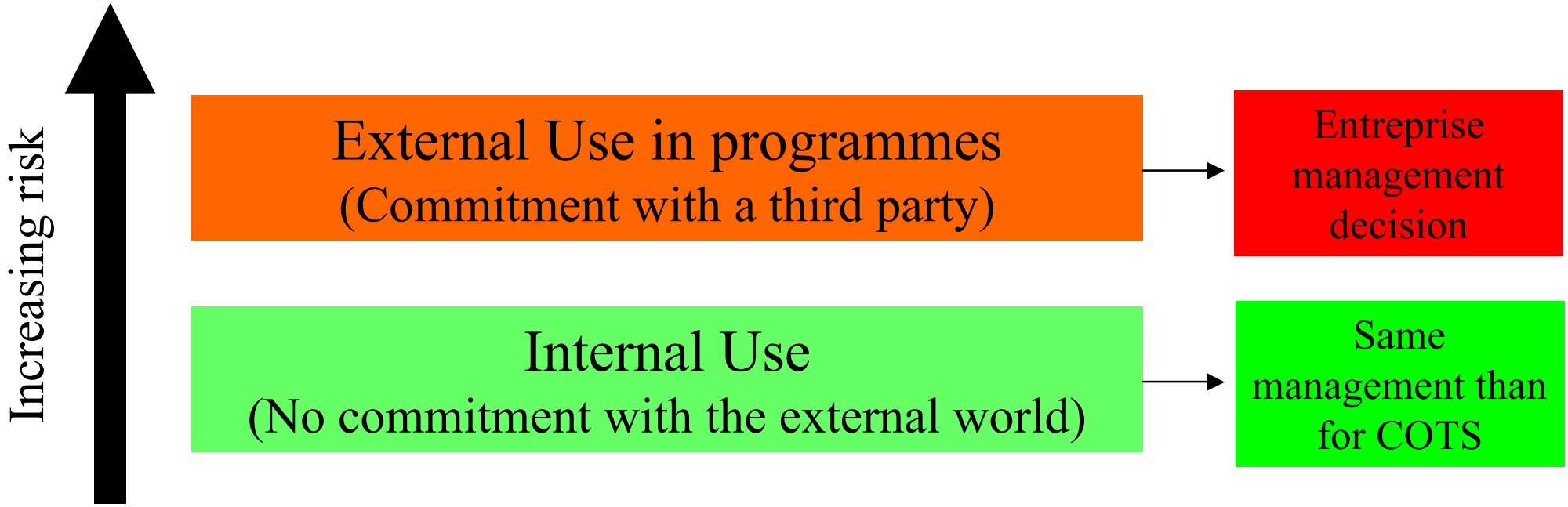
Identified risks (1/2)

- **Propagation of the terms of the license to our developments (GPL and similar licenses)**
  - Losing the ability to protect (patent, copyright) our software
  - Having to make available the sources of our developments
  - **SOLUTION :**
    - **Analysis of each major license**
    - **Intranet files** indicating how to deal which each major license
    - **Isolate Thomson-CSF developments** from freeware
    - Wherever its possible **provide freeware clearly separated** from Thomson-CSF developments
- **Configuration management**
  - Open source products evolve continuously
  - Having the source creates the ability to develop multiple variants
  - **SOLUTION :**
    - Appropriate infrastructure and regulations to **control the origin and the integrity** of the open source

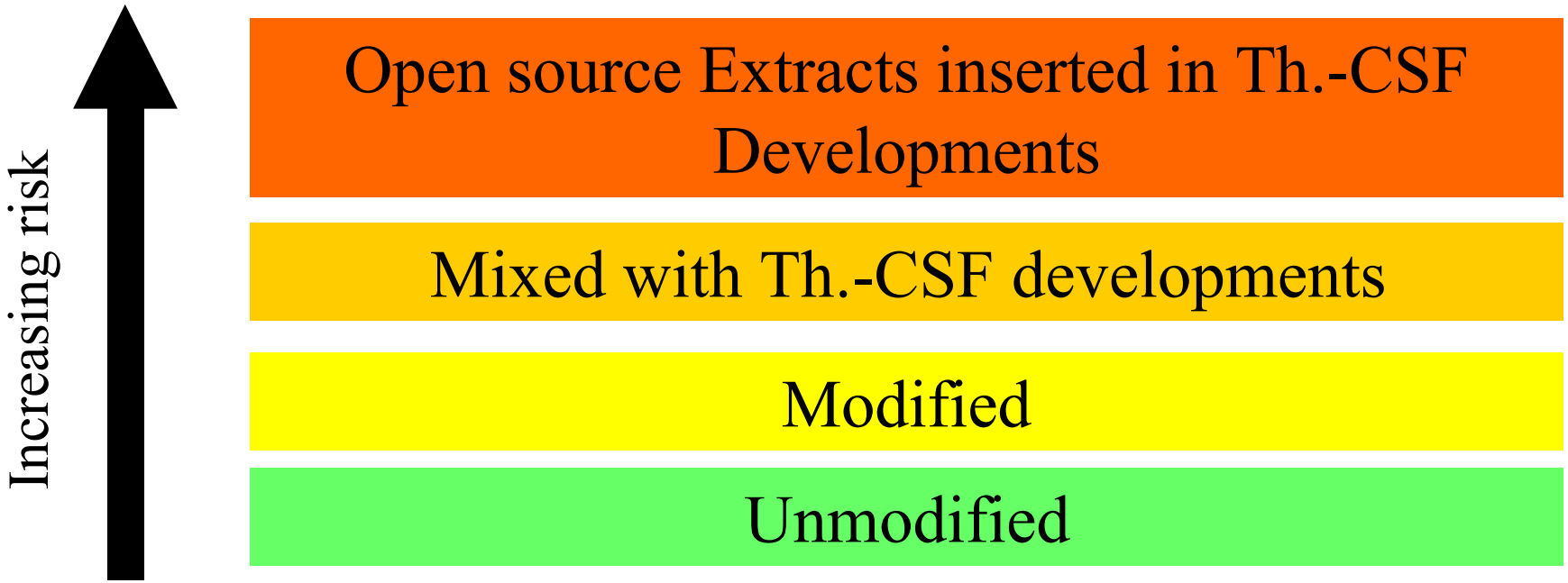
Identified risks (2/2)

- **Patented algorithms embedded into open source software**
  - **SOLUTION :**
    - Evaluate **potential risk (royalties) vs. advantages** of using the open source software
    - Consult the Web
- **Support**
  - Having the source doesn't give you automatically the skills
  - Who will be interested in the particular version you choose today 3 years from now ?
  - **SOLUTION :**
    - Create the **appropriate infrastructure**
    - Create the **appropriate skills**
    - Purchase the **appropriate support services**

Two major types of use of freeware in Thomson-CSF



Different ways to use open source

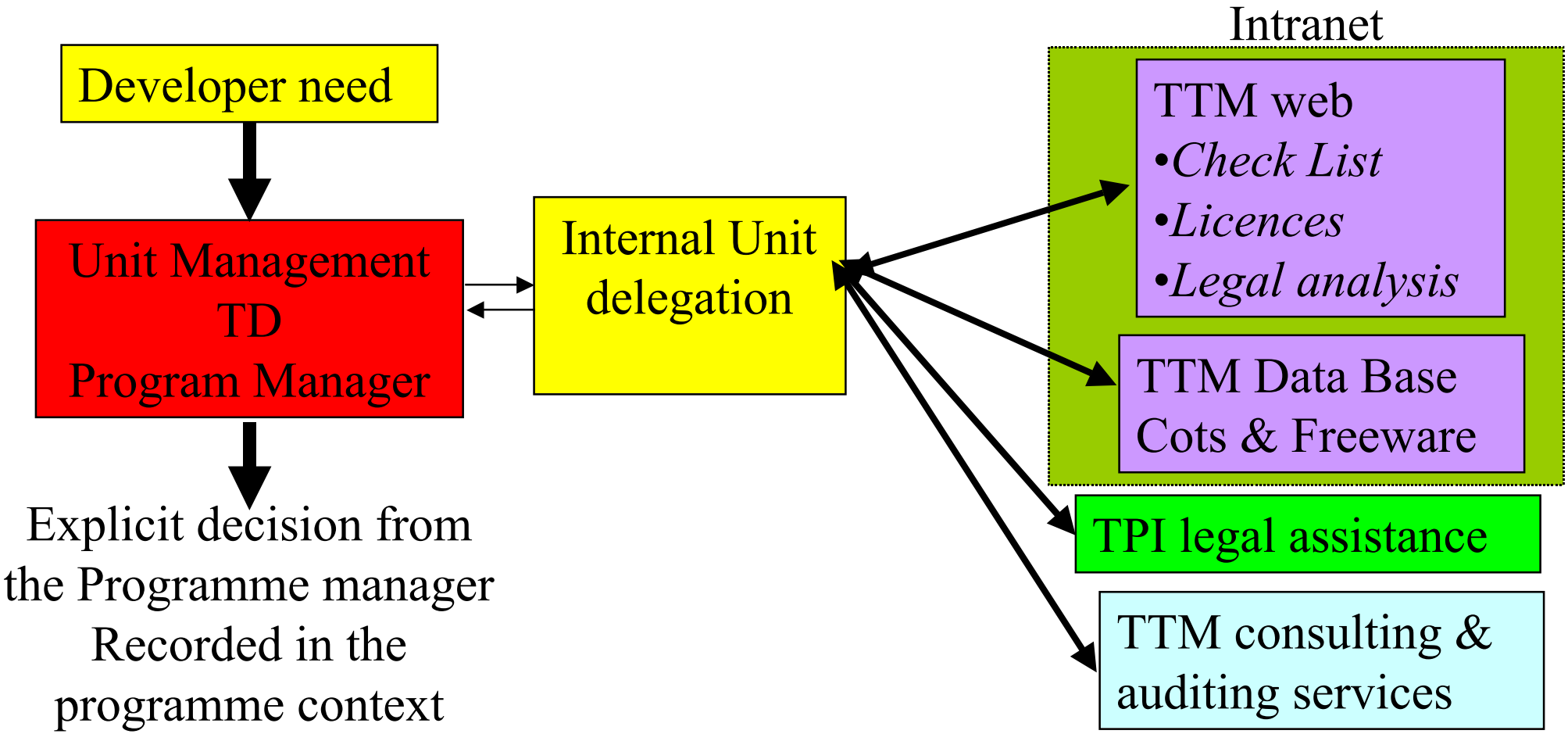


Proposed Thomson-CSF Policy

- **For Internal needs : Controlled usage centralised by each Unit**
  - **Control the integrity of the software (no day by day updates)**
    - Origin
    - Electronic Signature (whenever possible)
    - Internal repository
  - **Manage it just like software COTS :**
    - List of recommended products (as with COTS : avoid dispersion)
    - List of the free software used (track the use made with each software)

- **For systems delivered to customers : Explicit Authorization of the appropriate level of management in the Unit.**
  - **Controlled introduction in Thomson-CSF**
    - License vs. the type of use
    - Integrity of the software (origin, electronic signature, internal repository)
    - List of recommended products
    - List of the free software used

Proposed organization (use of open source on programmes)



Freeware Working Group follow on actions

- ***Rules of use of open source into Thomson-CSF (Q4 99)***
  - Group Directive (Q1 2000)
  - Implementation of the appropriate infrastructure by the Units (Q1 2000)
  - Definition of the appropriate actions to verify that the directive is correctly applied (Q1 2000)
  - Licenses analysis (Q1 2000)
    - Characteristics of the major open source licenses
- ***Linux (Mid 2000) : with the participation of Alcatel***
  - O.S.
  - Environment (tools, compilers, etc)
  - Support
- ***Policy to publish open source on the Internet (Mid 2000) : with the participation of Alcatel***

Open Source Software Policy

D. Potier

Qualification of « Open Source » Software Components

➔ **Definition: an "Open Source" software component is qualified if:**

- The **licence** protecting the component **has been analysed by TPI and documented.**
- **The usages** covered by the licence (used as such, modified and used, clearly separated, mixed with a Th.-CSF development, some extracts included in a Th.-CSF development, etc) **have been analysed and documented.**
- The **existence of third parties patents** protecting partly or completly the component **has been investigated** and documented.
- **An organization (internal or external) has been selected**, in charge of delivering the component (with the required level of integrity) and archiving versions of the components.

➔ **Responsabilities:**

- The Software CET, with the support of TPI, is in charge of qualifying "Open Source" software components, documenting and disseminating these informations *via* ThomWeb.

Internal use of « Open Source » (engineering env.)

➔ **Rules:**

- **Only qualified "Open Source"** software components can be integrated in engineering environments.
- **Standard Cots management rules,** in terms of evaluation, selection, recommendation, identification, configuration management, obsolescence, etc, apply to these components.

➔ **Responsabilities:**

- **The Engineering Manager and/or the Cots Manager** of the Entreprise are in charge of authorizing and managing "Open Source" software components used in the Entreprise engineering environments.

Use of « Open Source » in programmes (1)

➔ **Rules:**

- The integration in a programme of an "Open Source" software component is submitted to a **formal approval by the Programme Manager.**
- The approval process is comprized of **two phases:**
  - **Qualification** of the component.
  - **Check** of the qualifications against the **programme specific caracteristics** (customer, contracts, regulations, etc).
- The **formal approval** is required on a **programme per programme**, **component per component** basis.
- When approved:
  - The component is **managed according to standard Cots management rules**.
  - The component usage is **documented and traced.**

Use of « Open Source » in programmes (2)

➔ **Responsabilities:**

- At programme management level, the **Programme Manager** is responsible for the decision and its documentation.
- At entreprise level, the **Technical Director** is in charge of managing the approved "Open Source" Software Components and documenting and tracing their usage.
- At corporate level, the **Software CET and TPI** provide support on specific technical and legal issues related to the programme and the component.

Assessment of « Open Source » policy

➔ **Responsabilities:**

- The **Technical Director** of the Entreprise is responsible for **monitoring and assessing the "Open Source" software components policy in the Entreprise**.
- **The Software CET will assess the Entreprise "Open Source" software components policy** as part of the periodic Entreprise assessments regarding software policies.

Rules and duties
1/ Developer

➔ **You are not allowed to introduce open software in Thomson-CSF:**

- **Only the entity in charge of open source in your Unit can introduce "Open Source"**. This entity must make sure that this open source is taken from an identifeid source and hasn't been modified.
- You need to request from this Entity any new open source software
- If you use open source on programs an explicit authorization of the RCA is needed for EACH open source.
- This is true for extracts of open source used in Thomson-CSF developments

➔ **You are not allowed to publish open software outside Thomson-CSF:**

- **Only yout management can explicitely authorize the publication of software outside of the Group.**

Rules and duties
2/ RCA

➔ **You must**

- Check that the type of use made with the open source software is compatible with your program requirements and with the licence of the OSS.
- Explicitely declare every open source software used in your program:
- Remember that only qualified open source is authorized.
- Declare also extracts of open source used in Thomson-CSF
- Apply configuration management rules to open source software.

Rules and duties
3/ Entity in charge of Open Source

➔ **You must :**

- make sure that every new OSS in use in your Unit has been taken from a "clean" origin
- make sure that the licence analysis has been made
- avoid having different OSS for the same need
- decide which updates you introduce in your internal version and when.
- If the licence requires that any improvement made by Th.-CSF to the OSS must be published you must do these publications
- control the different versions of the original OSS that may appear in your Unit.
- Declare all the OSS used in your Unit to TTM
- Define the EXPERTISES and SKILLS (internal or external) needed.

Questions ?

For more information :

http://ttmweb.ttm.thomson-csf.com/itcp/linux/index.htm

Where freeware may not work (Freeware and patents)

- **X11 and the X Consortium**
  - XOR on a bitmap memory (cursor handling) in X11.
  - Major vendors providing X11 had to pay royalties
- **LZW and UNISYS**
  - LZW is a UNISYS patent
  - UNISYS now wants to oblige all the non authorised users of LZW to pay a license.