



A PUBLIC OPEN-SOURCE ENVIRONMENT FOR A SAFER INTERNET ACCESS

WWW.POESIA-FILTER.ORG



Information Society



Poesia software architecture



Poesia requirements

- **Classroom traffic filtering**
- **No configuration on filtered workstations**
- **No mean to avoid the filter**



Development context

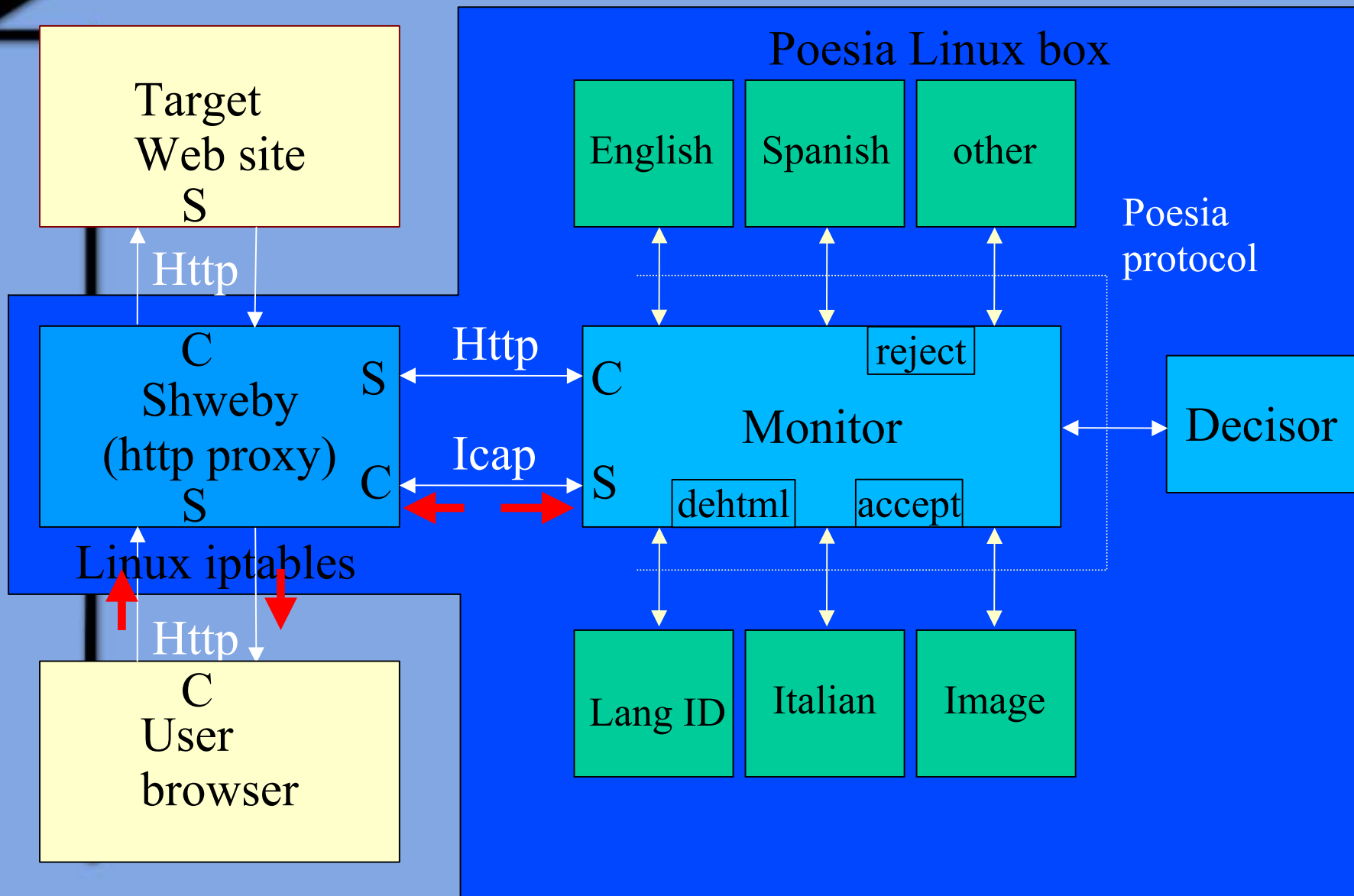
- **Open-source development**
- **Linux**
- **EEC Safer Internet Action Plan**



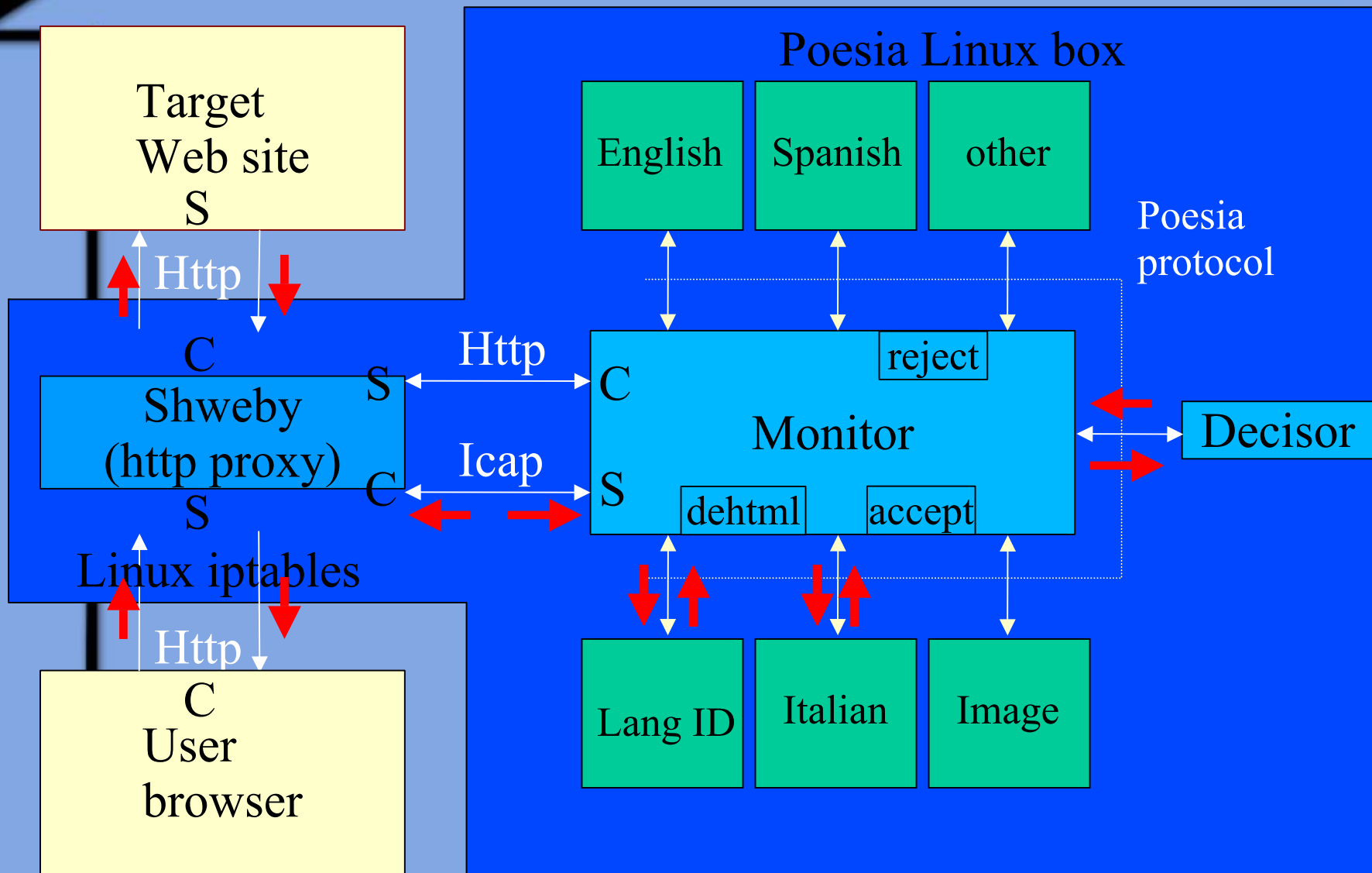
Development constraints

- Collaborative distributed devpt
- Different devpt cultures
 - Natural Language Processing experts
 - Image experts
 - Existing source code base
 - Different “prefered” language
- Internet standards compliance

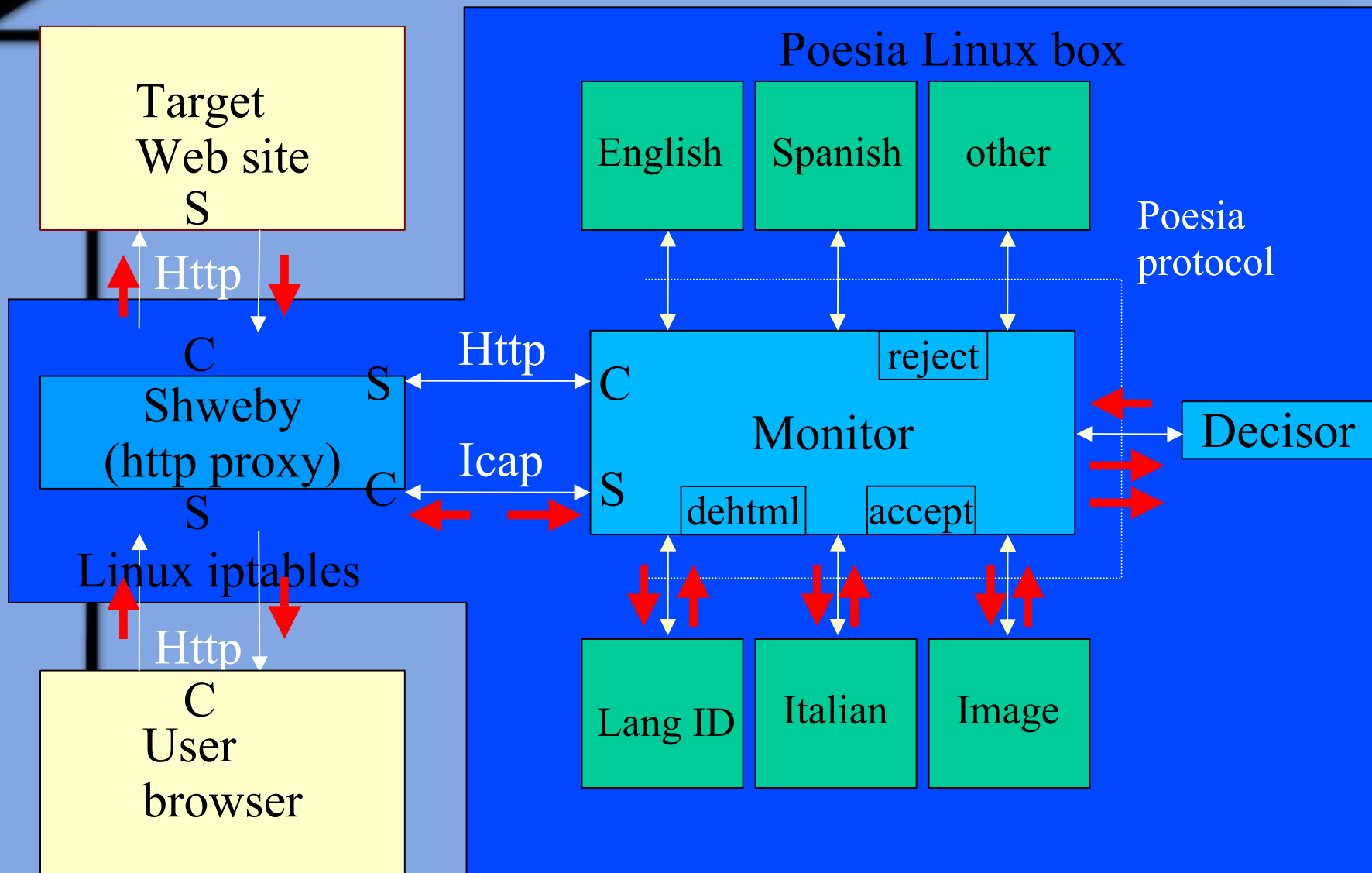
Poesia architecture diagram



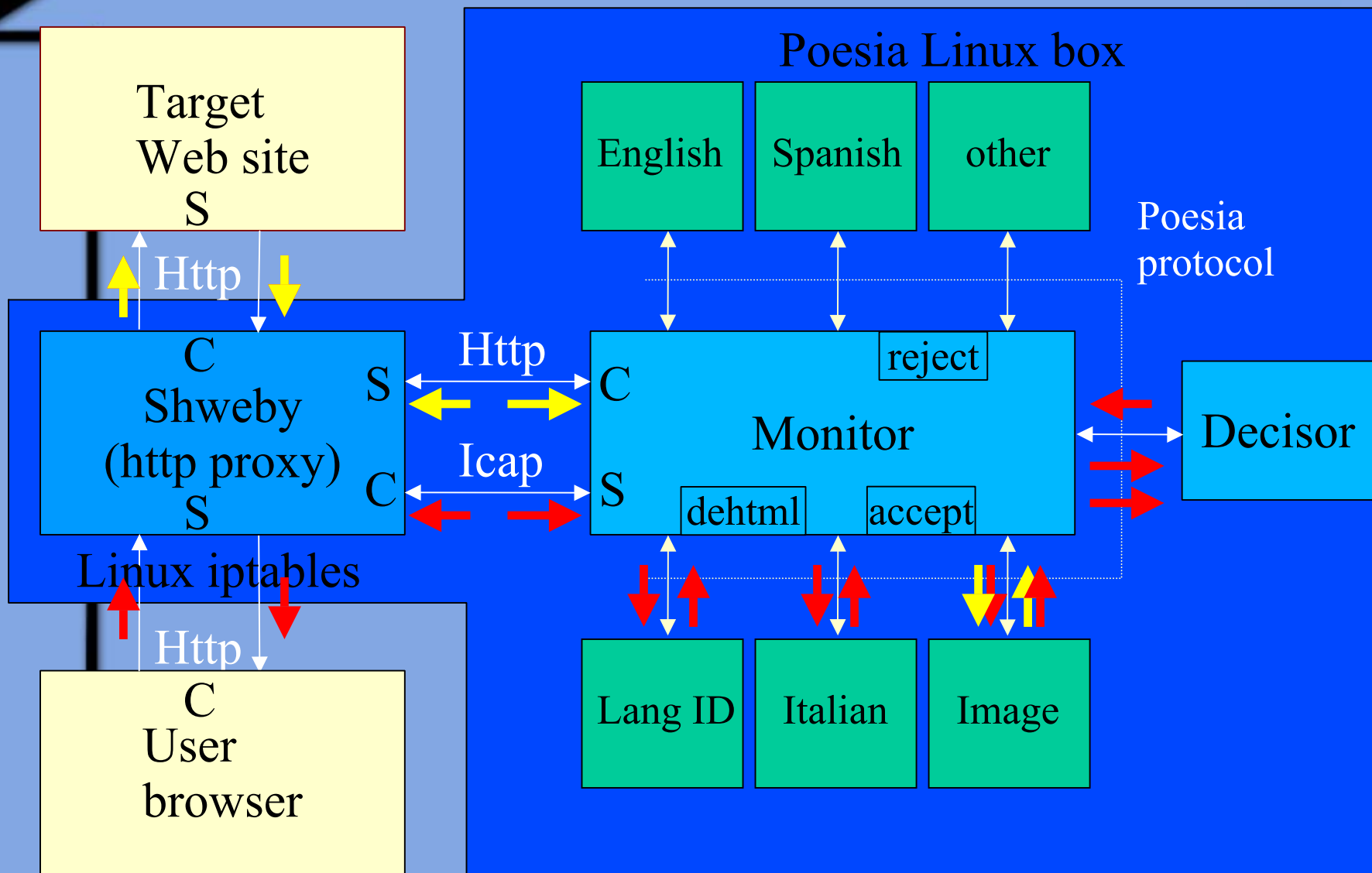
Poesia architecture diagram



Poesia architecture diagram



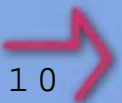
Poesia architecture diagram





Technical choices

- **Modularity : 1 kind of filter : 1 executable**
- **Hub architecture**
- **Communications**
 - **Message based**
 - **Monitor <-> Filter only**
 - **Unix pipes**
- **Central decision mechanism**
- **ICAP compliance**





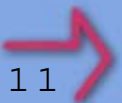
Message-based communication

Pros

- Low coupling
- Independance / programming language
- Debug text-based protocol
- Pipe performance

Cons

- Loose type checking
- Finally, not the announced Babel tower
- Parsing and message handling overhead
- No possible machine distribution





Modularity

- More different natural languages
- Other kind of filters ?

Reusability

- Decisor, Image and Spanish filter use common base classes
- Common poesia protocol handling library
- Reusable framework for new filters



Return on experience (about software engineering)

Positive

→ Collaborative development

→ hierarchy of makefiles

→ cvs

→ Modular architectures

→ Common library and framework

Improvements

- Earlier integration with « empty shell » modules

- More automated unit and integration testing

- Less different technologies

- More technology recommendations