

# Ologid

A “DBus-enabled logic inference server”

**FSO|SHR '09**  
Users & Developers Convention



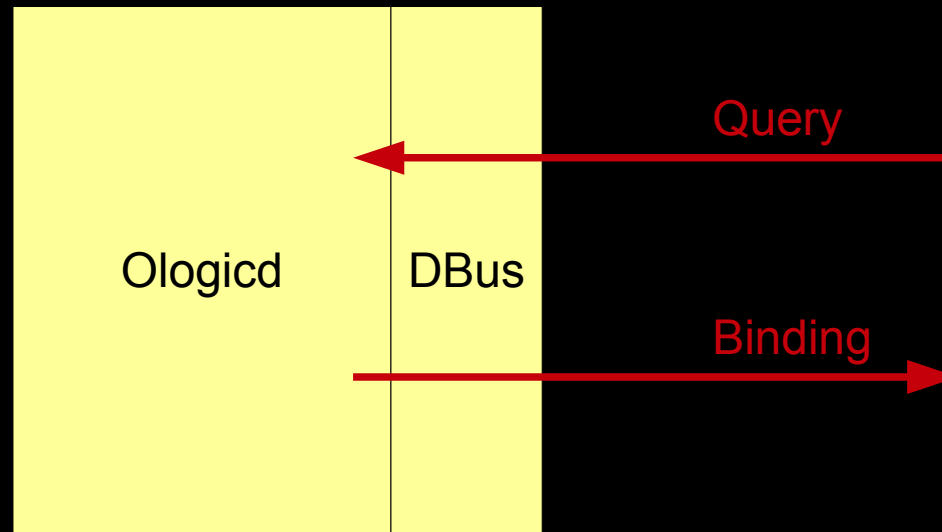
# Ologid – Summary

- The initial idea
- Inference use cases
- About XSB
- About Flora-2
- Architecture
- Expected logic frameworks
- Current state
- Conclusion

# Ologicd – The initial idea

## “DBus logic inference server” (1/3)

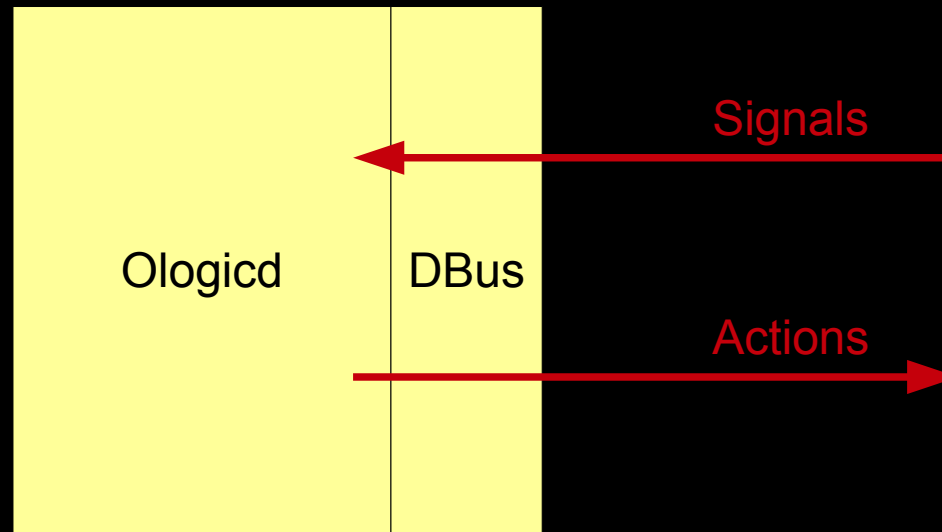
- Serves deductions
  - Give it a query formula, it returns its truth conditions
  - This may imply querying additional DBus servers



# Ologicd – The initial idea

## “DBus logic inference server” (2/3)

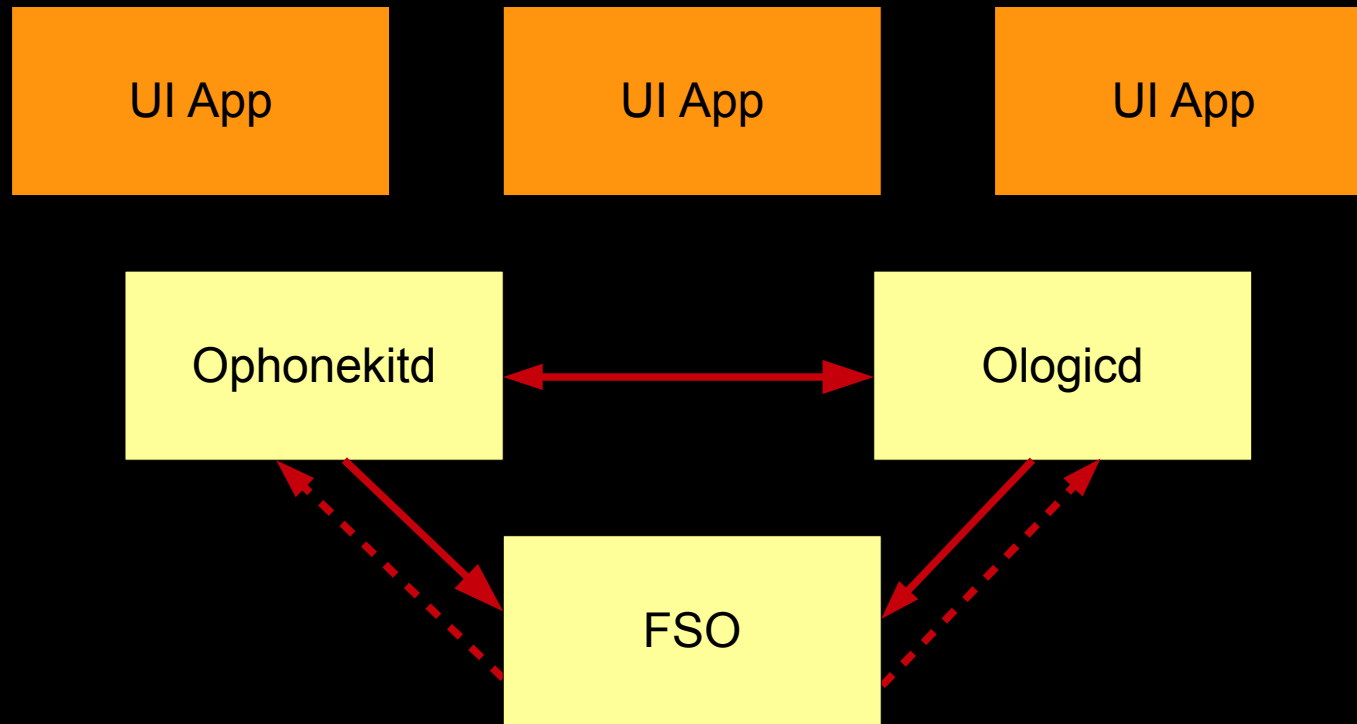
- Stays aware of system state changes
  - Registers to signals, and apply a specific query
  - May imply triggering actions, or firing other signals



# Ologicd – The initial idea

## “DBus logic inference server” (3/3)

- Collaborator in a DBus network
  - Its clients are UI apps and ophonekitd
  - It is a client of frameworkd and ophonekitd
  - ophonekitd and ologicd may be mutual clients



# Ologica – The initial idea Agents

- Domain-specific inference sources
- Autonomous execution (in its own thread)
- May collaborate with other agents
- Perfectly maps to the Component concept

# Ologid – Inference use cases

## Data mediation

- Front-end over opimd
  - Managing complex queries in one method call
- Filtering content in some situations
  - Use context to serve adhoc content
    - profile, activity, location, ...
  - May help in better completion and “smart” views
- Data reconciliation
  - Integrating|correlating data from multiple sources
    - SIM, desktop PIM software, received mails and SMS

# Ologid – Inference use cases

## Rule processing

- Modifying phone behaviour
  - Use context to react differently
    - profile, activity, location, ...
  - May help in call protocol selection
- Using call patterns
  - Recognize contacts and call differently
    - “Beep me, I'll call you free of charge”

# Ologid – About XSB

## Overview

- Prolog interpreter
- Tabled predicates
- Multi-thread engine
- Provides loads of Prolog modules
  - Database access/mapping, ...
- Funky build system
  - Automake but not Autoconf (but very portable)
- No shared library
  - Provides an xsb.o

# Ologid – About Flora-2 Overview

- Frame Logic engine
- Syntax very close to standard F-Logic
- Implemented in Prolog/XSB
- Transaction Logic support

# Ologid – About Flora-2 Frame Logic

- Brings object-oriented paradigm to logic
- Supports
  - Object identity
  - Inheritance
  - Polymorphism
  - Encapsulation
- Declarative object-oriented formalism

# Ologid – About Flora-2

## Some examples

```
John[name → 'John Doe', phones → {1234567, 76442211},  
  children → {Bob, Mary}]
```

- Structures can be nested

```
Jane[spouse → John[address → '123 Main st.']]
```

- Class membership with :
- Subclass relationship with ::

```
John:Person.  
Alice:Student.  
Student::Person.
```

```
Person:EntityType.  
Student:EntityType.
```

# Ologid – About Flora-2

## Some examples

- Variables starts with ?
- Method definitions

```
?P[ageAsOf(?Year) → ?Age] :-  
  ?P:Person, ?P[born → ?B], ?Age is ?Year-?B.
```

- Attributes are methods without arguments
- Queries

```
?- John[born → ?Y, children → ?C], ?C[born → ?B], ?B > ?Y+30.
```

```
?- John[ageAsOf(?Y) → 30, children → ?C], ?C[born → ?B], ?B > ?Y.
```

# Ologid – About Flora-2

## Transaction Logic

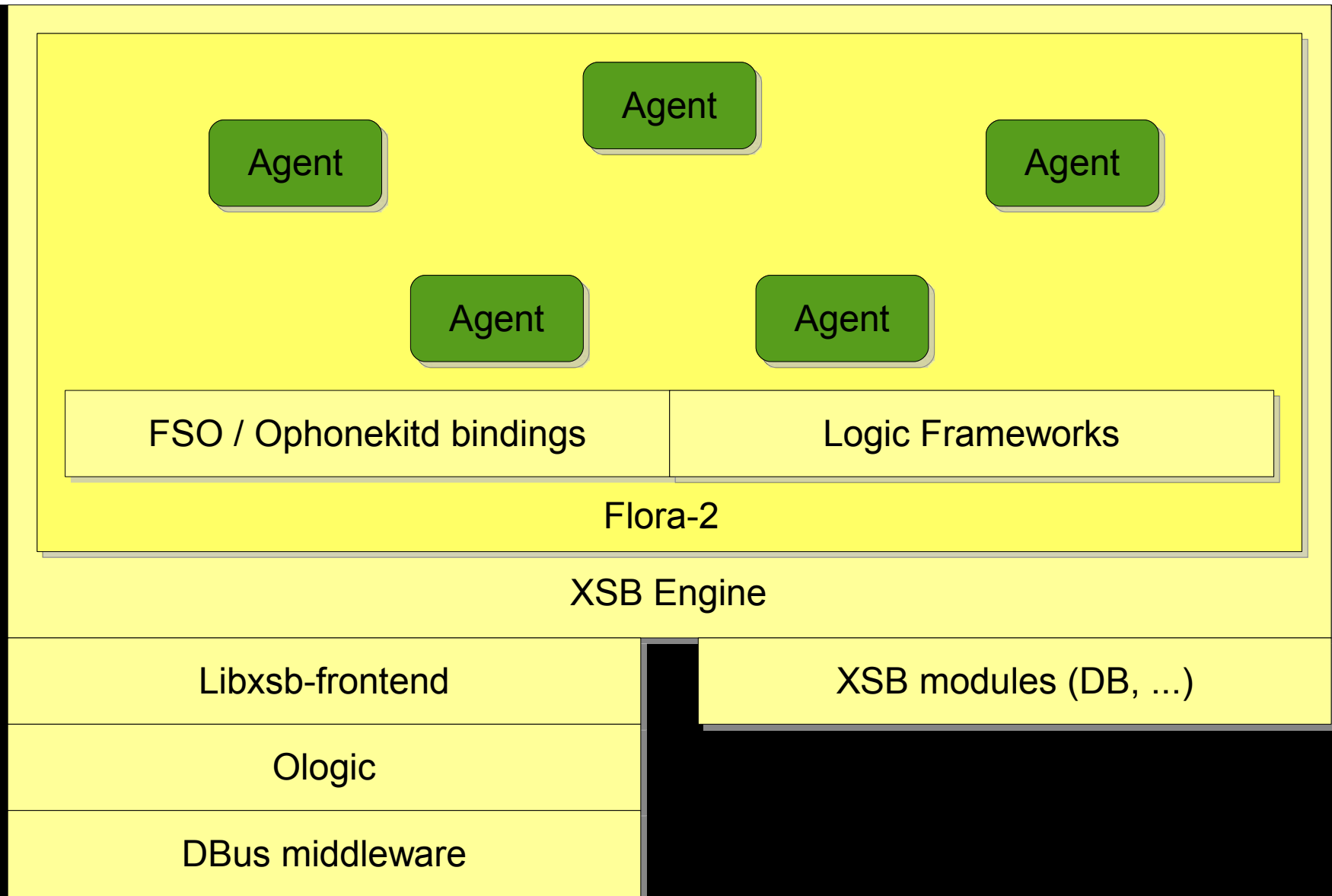
- Updates in usual logic programming
  - `assert/retract` have no logical semantics
  - Can't be backtracked

?- `assert(p), q.`

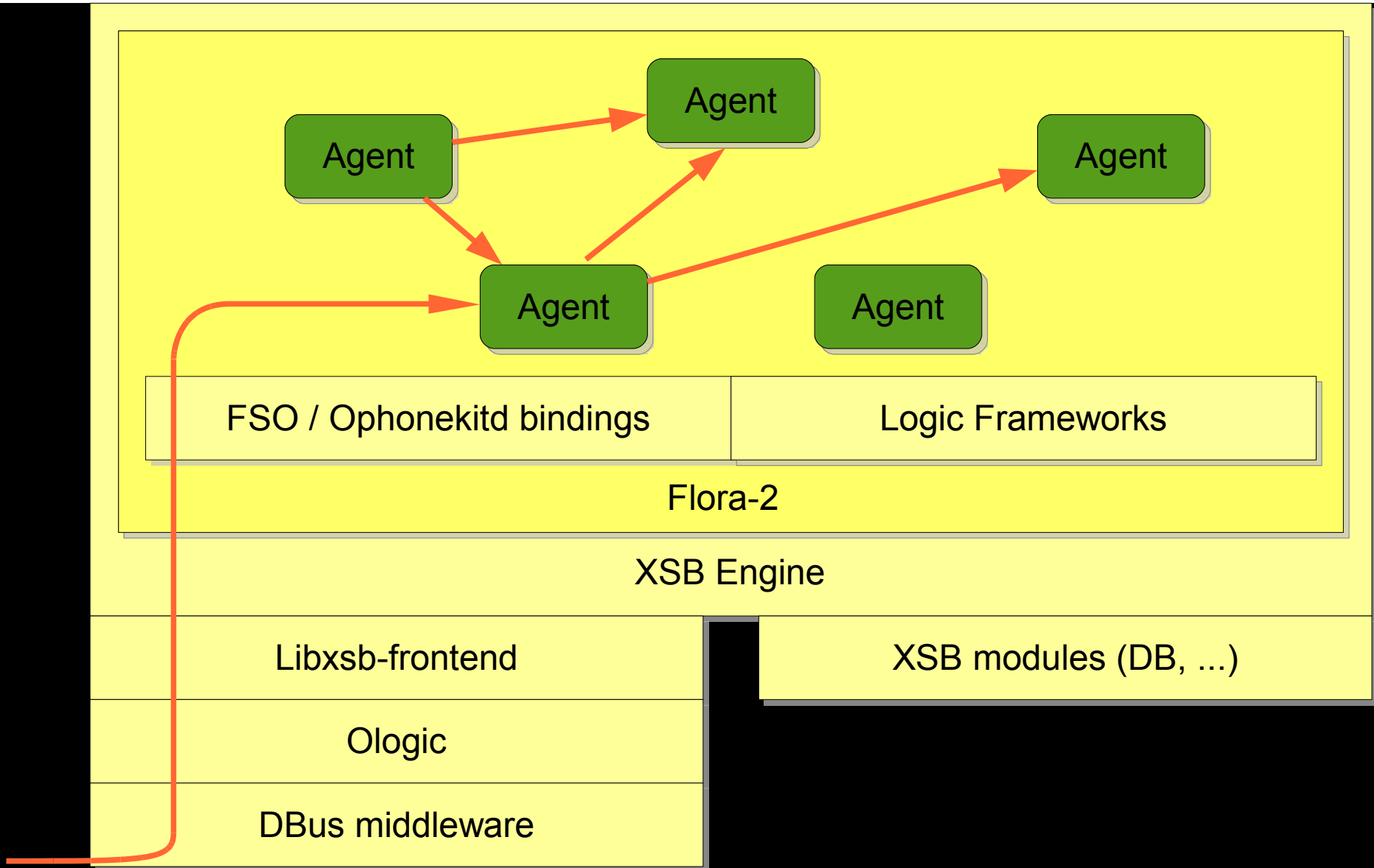
– If `q` is false, `p` stays.

- Transaction Logic
  - `t_insert{fact}/t_delete{fact}` are transacted
  - Can be backtracked

# Ologic – Architecture Overview



# Ologic – Architecture Overview



# Ologid – Architecture

## Libxsb-frontend

- Vala-oriented C wrapper
- Full management of XSB threads
- Full manipulation of Prolog terms
- Encapsulate xsb.o

# Ologid – Architecture

## Access to phone features

- Needs access to UI phone features
  - Rejecting and ignoring a call
  - Routing a call to another destination
  - ...

# Ologid – Logic Frameworks

## Web crawling

- Crawls the web / semantic web
  - Pull RDF data
  - Extract from HTML pages
- Gives more facts to deduce interesting things
- Typical use case
  - Pizzerias around

# Ologid – Logic Frameworks

## Trust network

- User specify how much he trusts some others
- Others provide agents
- Others provides rule and/or fact sets
- Others provides their own trust assertions
- This builds a trust network
- Typical use case
  - Best pizzerias around

# Ologica – Logic Frameworks

## Proof builder

- Usually, give a query and get some bindings
- Proof Builder
  - Give the query and resulting bindings
  - Get the tree of rules to go from query to bindings
- Typical use case
  - Best pizzeria around

# Ologid – Current state

- Done
  - Libxsb-frontend
  - Ologid
- To do
  - Bind FSO / Ophonekitd APIs in Flora-2
  - Wrap Flora-defined agents as Modulo components
  - Implement the Logic Frameworks

# Ological – Conclusion

# Ologid – Resources

- XSB:  
<http://xsb.sourceforge.net>
- Flora-2:  
<http://flora.sourceforge.net>
- Libxsb-frontend:  
<http://git.shr-project.org/git/?p=libxsb-frontend.git>
- Ologid:  
<http://trac.shr-project.org/trac/wiki/Ologid>
- Frame Logic:  
<http://flora.sourceforge.net/aboutTR.php>
- Transaction Logic:  
<http://flora.sourceforge.net/aboutTR.php>