Disko-
Embedded Linux UI Frame Work

Siji Sunny
siji.sunny@exnxt.com
www.sijisunny.com

exnxt

OverView

- History of Embedded System UI's
- Challenges in embedded system design
- Disko- Features & Frame work
- Examples

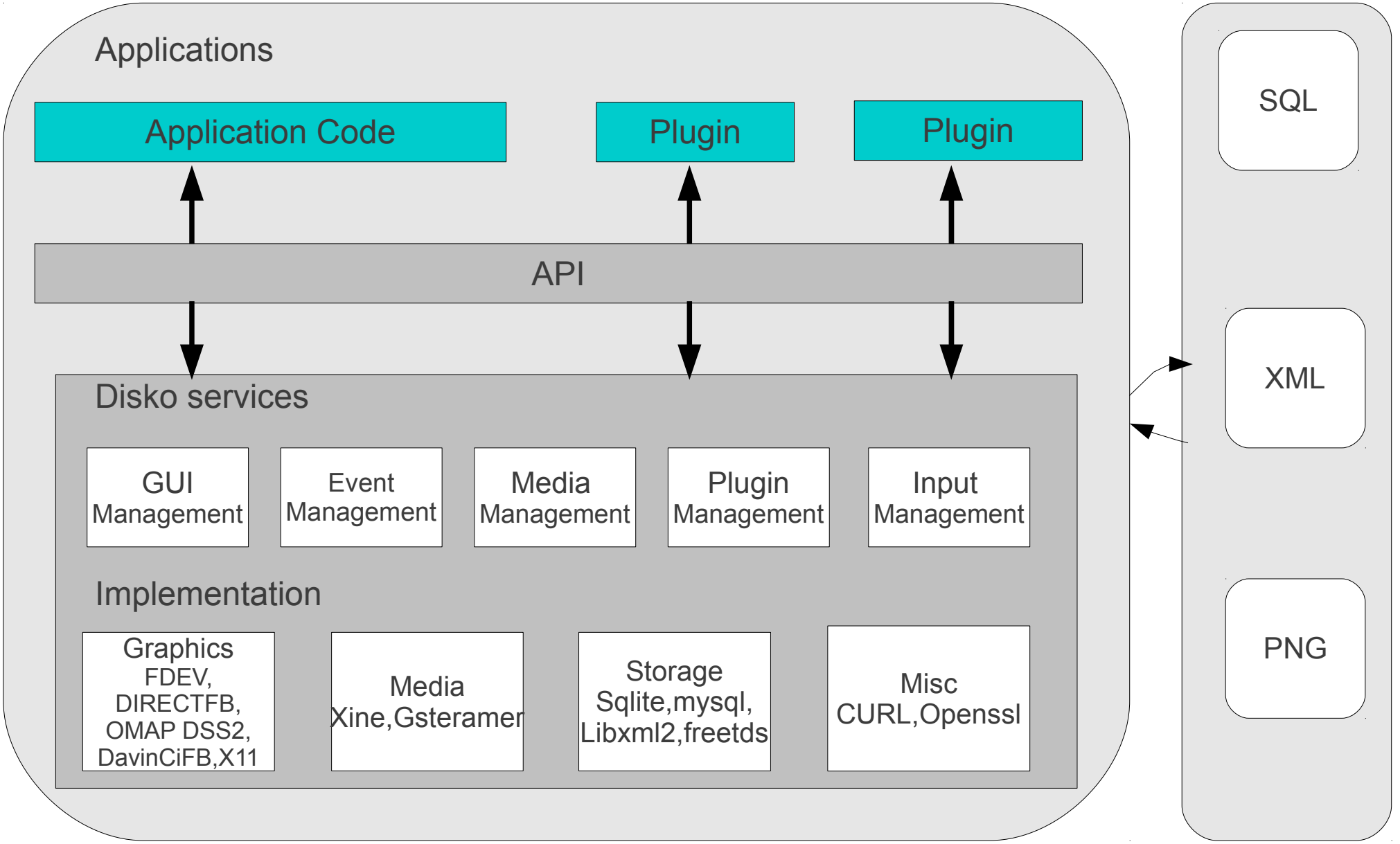
Challenges

- CPU -Architecture Difference
- Cross Compilation Complexities
- Interoperability
- Multiple Standards
- Delivered by multiple vendors

Disko- In general

- Simplified frame work for complex and flexible UI development
- Developed in C++
- Offers the runtime implementaiton of basic libraries
- Less dependencies
- Support for various graphical environments
  X11, Framebuffer, DirectFb
- API provides, both level and high level acess.

Application Architecture



GUI Management- Using Windows

- Root Window
- Main Window
- Popup Window

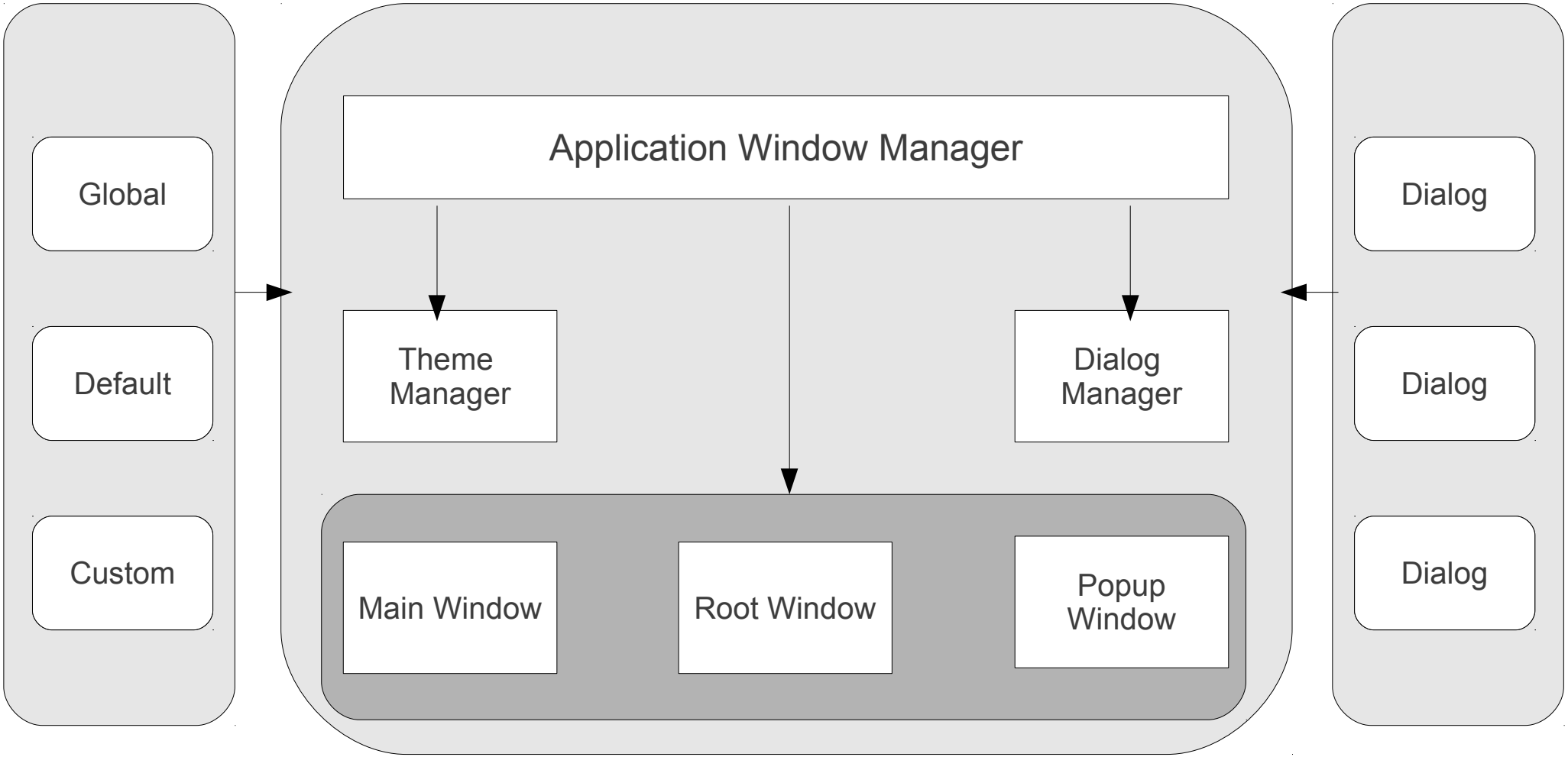
Dialog Management

- Dialog files store information about layout and structure of window content
- It is the easiest way of creating content
- It frees from programming a fixed dialog structure .

Plugin Management

- Share code between several different application
- Disko stores it plugin informations within an sqlite database
- Database contains the filename and location of a plugin as well as plugin specific parameters

GUI Components



Disko-Examples

Descriptions

- FreeTDS

Is a set of libraries for Unix and Linux that allows programs to natively talk to Microsoft SQL Server and Sybase databases

- cURL

- **DirectFB**

Is a software library for GNU/Linux/Unix-based operating systems that provides graphics acceleration, input device handling and abstraction layer, and integrated windowing system with support for translucent windows and multiple display layers on top of the Linux framebuffer without requiring any kernel modifications

Questions

Thank You

www.sijisunny.com

www.diskohq.com