



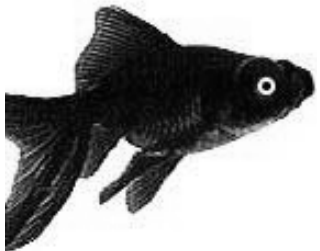
Fauna spotting in the DiSy menagerie

A guide to software projects in DiSy

Ben Leslie

benjl@cse.unsw.edu.au

October 2004



SOFTWARE PROJECTS

Kenge

- Libraries: data structures, drivers, `libc`, L4 convenience libraries
- Build system: generate boot images for different platforms
- Tools: `dite`, Magpie IDL compiler, `dd_ds1.py`

SOFTWARE PROJECTS

Kenge

- Libraries: data structures, drivers, `libc`, L4 convenience libraries
- Build system: generate boot images for different platforms
- Tools: `dite`, Magpie IDL compiler, `dd_ds1.py`

Iguana

- An L4 based OS “personality”
- Designed for embedded systems
- Designed for running legacy systems
- Mungi derivative

SOFTWARE PROJECTS

Kenge

- Libraries: data structures, drivers, `libc`, L4 convenience libraries
- Build system: generate boot images for different platforms
- Tools: `dite`, Magpie IDL compiler, `dd_ds1.py`

Iguana

- An L4 based OS “personality”
- Designed for embedded systems
- Designed for running legacy systems
- Mungi derivative

Wombat

- An port of Linux 2.6 to Iguana
- Portable approach

IGUANA

This will be a *brief* overview of Iguana

More info: <http://www.disy.cse.unsw.edu.au/Iguana/>

Why??

- Started as an idea on the bus back from Linux.conf.au 2004
- Further developed due to customer desire to run Wombat next to proprietary software.
- Primarily *engineering* not *research*



IGUANA

This will be a *brief* overview of Iguana

More info: <http://www.disy.cse.unsw.edu.au/Iguana/>

Why??

- Started as an idea on the bus back from Linux.conf.au 2004
- Further developed due to customer desire to run Wombat next to proprietary software.
- Primarily *engineering* not *research*

What does Iguana provide?

- Convenient way of using L4 primitives
- A flexible capability based framework for protection
- Virtual memory management
- Support for user level components

IGUANA

This will be a *brief* overview of Iguana

More info: <http://www.disy.cse.unsw.edu.au/Iguana/>

Why??

- Started as an idea on the bus back from Linux.conf.au 2004
- Further developed due to customer desire to run Wombat next to proprietary software.
- Primarily *engineering* not *research*

What does Iguana provide?

- Convenient way of using L4 primitives
- A flexible capability based framework for protection
- Virtual memory management
- Support for user level components

Iguana's not Mungi ... but it borrows a lot of ideas (and code) from Mungi.

SINGLE ADDRESS SPACE (SAS)

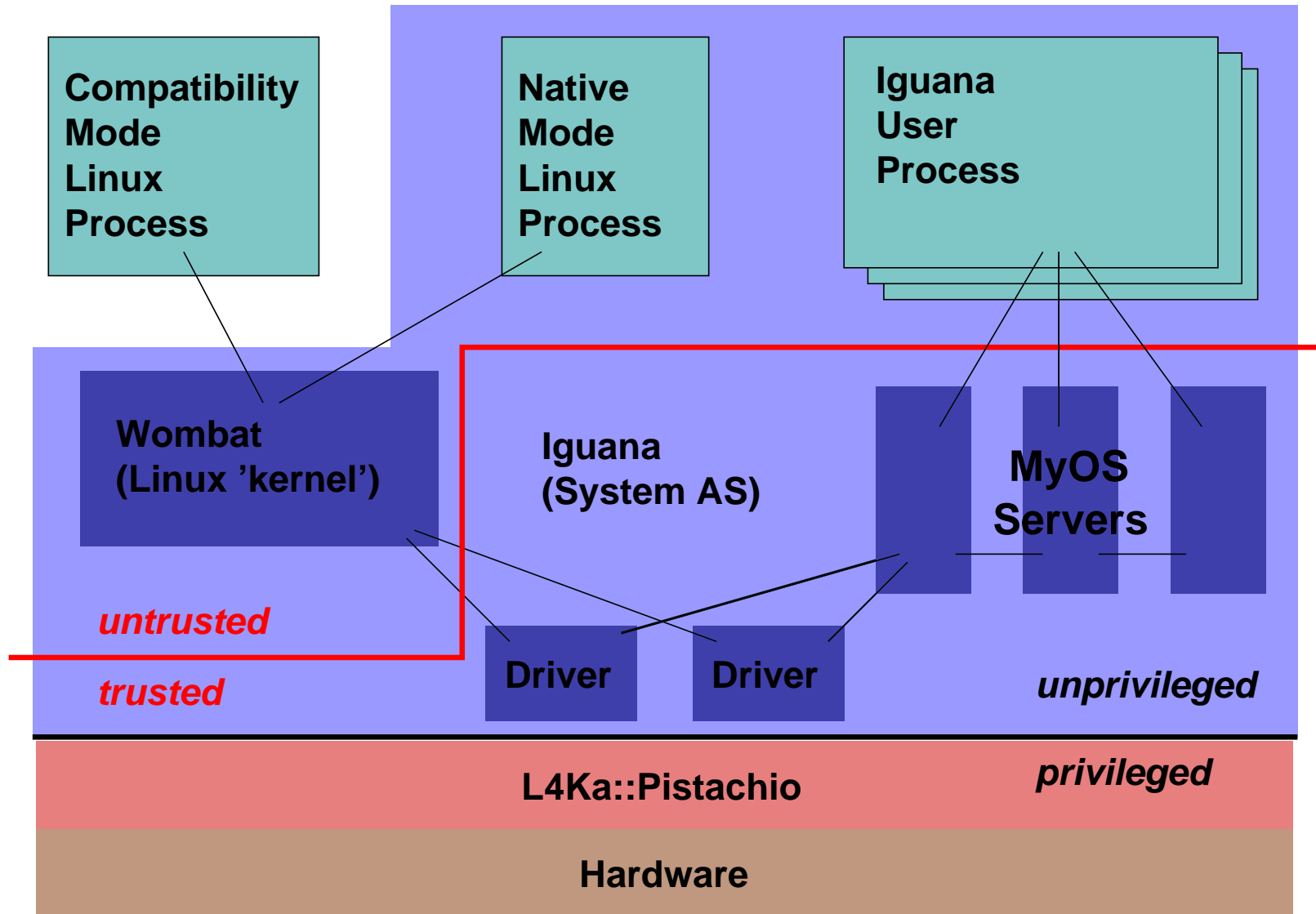
- eases sharing of data
- accessed enforced by protection domains
- SAS layout supports fast-address-space switching on ARM

SINGLE ADDRESS SPACE (SAS)

- eases sharing of data
- accessed enforced by protection domains
- SAS layout supports fast-address-space switching on ARM

Iguana's not Mungi ... because the SAS is not pure,
persistent or distributed

EXAMPLE



OBJECTS

WARNING: Don't confuse with Mungi objects

OBJECTS

WARNING: Don't confuse with Mungi objects

“a unique concrete instance of an abstract data type (that is, a conceptual structure including both data and the methods to access it) whose identity is separate from that of other objects”

OBJECTS

WARNING: Don't confuse with Mungi objects

“a unique concrete instance of an abstract data type (that is, a conceptual structure including both data and the methods to access it) whose identity is separate from that of other objects”

- Objects have a unique object ID (OID) – address of instance data in the SAS.
- Objects have methods that can be invoked.
- Methods are grouped into interfaces.
- Interfaces have a unique interface ID. consisting of the OID + interface number.

Object are very similar to “component-instances” in the Mungi Component System (MCS).

OBJECTS

WARNING: Don't confuse with Mungi objects

“a unique concrete instance of an abstract data type (that is, a conceptual structure including both data and the methods to access it) whose identity is separate from that of other objects”

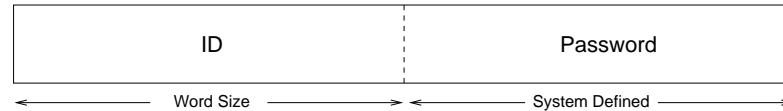
- Objects have a unique object ID (OID) – address of instance data in the SAS.
- Objects have methods that can be invoked.
- Methods are grouped into interfaces.
- Interfaces have a unique interface ID. consisting of the OID + interface number.

Object are very similar to “component-instances” in the Mungi Component System (MCS).

Iguana's not Mungi ... because objects are directly supported by the system

CAPABILITIES

A token that confers some access rights.

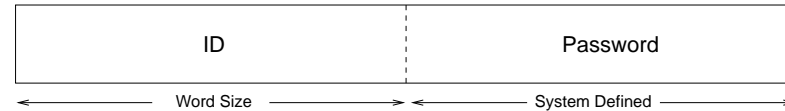


Iguana has two different kinds of capabilities:

- master capability
 - ID is the Object ID.
 - created during object creation
 - allows creation of further caps
 - confers rights on all methods of the object
- invocation capability
 - ID is an Interface ID.
 - created by someone with the master capability
 - confers rights to invoke methods on a single interface

CAPABILITIES

A token that confers some access rights.



Iguana has two different kinds of capabilities:

- master capability
 - ID is the Object ID.
 - created during object creation
 - allows creation of further caps
 - confers rights on all methods of the object
- invocation capability
 - ID is an Interface ID.
 - created by someone with the master capability
 - confers rights to invoke methods on a single interface

Iguana's not Mungi ... because caps on interfaces are support by the system, not middleware

SYSTEM OBJECTS

- Protection domains
- Memory sections
- Threads
- Sessions
- Hardware
- External address spaces

SYSTEM OBJECTS

- Protection domains
- Memory sections
- Threads
- Sessions
- Hardware
- External address spaces

Iguana's not Mungi ... because accessing system services is done through the same invocation model as accessing other services

PROTECTION DOMAINS

All code executes within a protection domain.

Capabilities are stored in a two-level data structure:

- Protected array of *Clists*
- Clists are user accessible list of capabilities

PROTECTION DOMAINS

All code executes within a protection domain.

Capabilities are stored in a two-level data structure:

- Protected array of *Clists*
- Clists are user accessible list of capabilities

Each system object is owned by a protection domain. When a protection domain is destroyed so are all its system objects.

Iguana's not Mungi ... because protection domains are all created explicitly

THREADS

- Created within a protection domain
- Are a way for managing L4 threads – not a different primitive
- Methods for creating, starting, destroying threads

THREADS

- Created within a protection domain
- Are a way for managing L4 threads – not a different primitive
- Methods for creating, starting, destroying threads

Iguana's not Mungi ... because threads don't migrate between protection domains

MEMORY SECTIONS

- Represent area of virtual memory – similar to Mungi objects
- Objects reside within memory sections
- Has a set of “pseudo-methods” for read, write and execute.
- Have an associated server.

MEMORY SECTIONS

- Represent area of virtual memory – similar to Mungi objects
- Objects reside within memory sections
- Has a set of “pseudo-methods” for read, write and execute.
- Have an associated server.

Iguana's not Mungi ... because faults are treated as pseudo-method calls

SESSIONS

- Used to establish a communication channel between a client and a server.
- Used to establish a shared memory region and clist.
- Implicit session between all threads and the Iguana server.

SESSIONS

- Used to establish a communication channel between a client and a server.
- Used to establish a shared memory region and clist.
- Implicit session between all threads and the Iguana server.

Iguana's not Mungi ... because it uses the words client and server

EXTERNAL ADDRESS SPACES

- Provide raw L4 address space that are not part of the SAS.
- Provided to allow implementation of, for example, Wombat.
- Are “second class citizens” – don’t hold capabilities, can’t access normal services

EXTERNAL ADDRESS SPACES

- Provide raw L4 address space that are not part of the SAS.
- Provided to allow implementation of, for example, Wombat.
- Are “second class citizens” – don’t hold capabilities, can’t access normal services

Iguana’s not Mungi ... because it breaks the pure single address space model