XCL: An Xlib Compatibility Layer for XCB

Jamey Sharp Bart Massey
Computer Science Department
Portland State University
Portland, Oregon USA 97207–0751
{jamey,bart}@cs.pdx.edu

http://xcb.cs.pdx.edu/

Overview

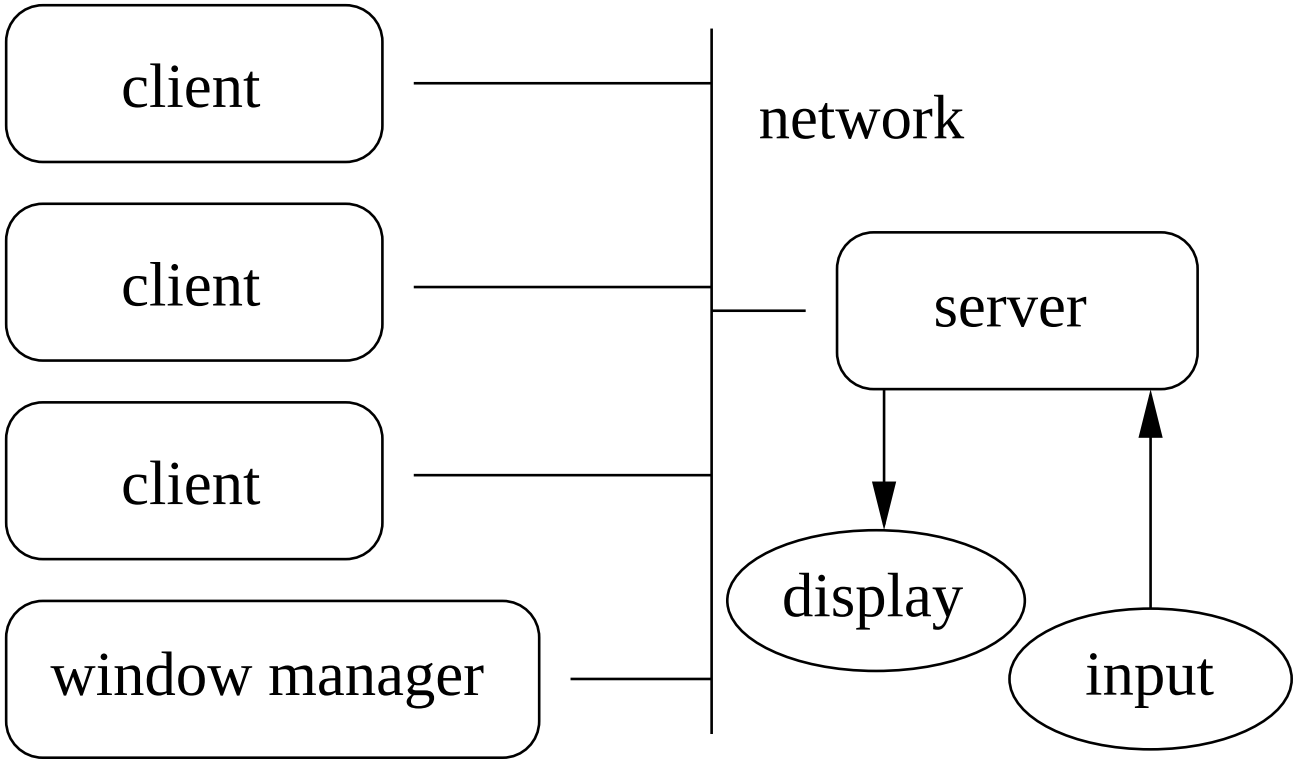
- The X Window System
- XCB
- XCL
- Summary

Thesis

Application of modern tools and X experience to X Window System client libraries can produce modern code

- by separation of concerns
- by design of new interfaces
- by re-implementing existing interfaces

What is X?

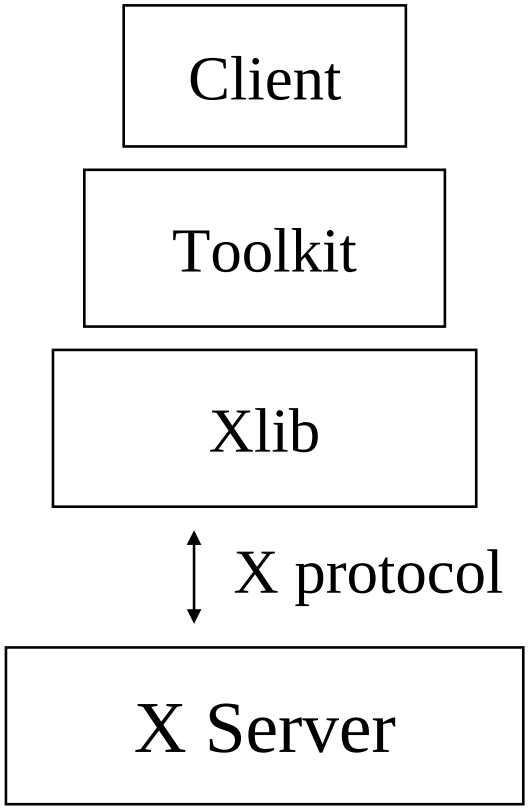


X Window System Architecture

X Strengths

- well designed
- separates policy from mechanism
  - window managers
  - toolkits
- network protocol specification separates hardware from apps
  - servers from XFree86, Sun, others
  - client libraries for many languages and platforms
  - client and server implementations are interoperable

Xlib: C Protocol Binding



Xlib's role in the X Window System

Motivations for XCB

Why change the Xlib API?

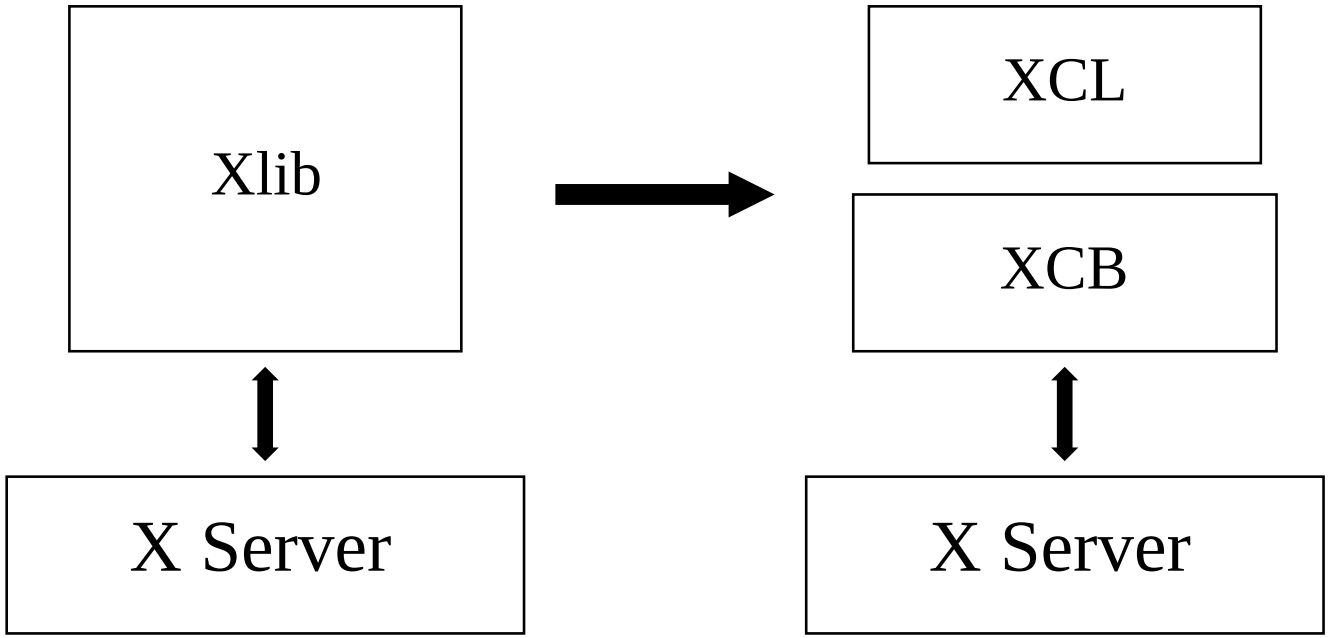
New API enables a light-weight, flexible implementation style.

- Smaller and simpler API and implementation
- Some optimizations affect API requirements
  - Latency is a problem for existing X applications
  - Xlib API supports latency hiding in special cases
  - XCB API supports latency hiding in all cases
- Support both single and multi-threaded clients with one API (c.f. Massey, "X meets Z")

XCB Evaluation

- most planned features implemented
- expected benefits have been successfully achieved
  - code size is particularly good: 27kB
- concept has been proven: modern implementation with new API

XCL: an Xlib Compatibility Layer for XCB

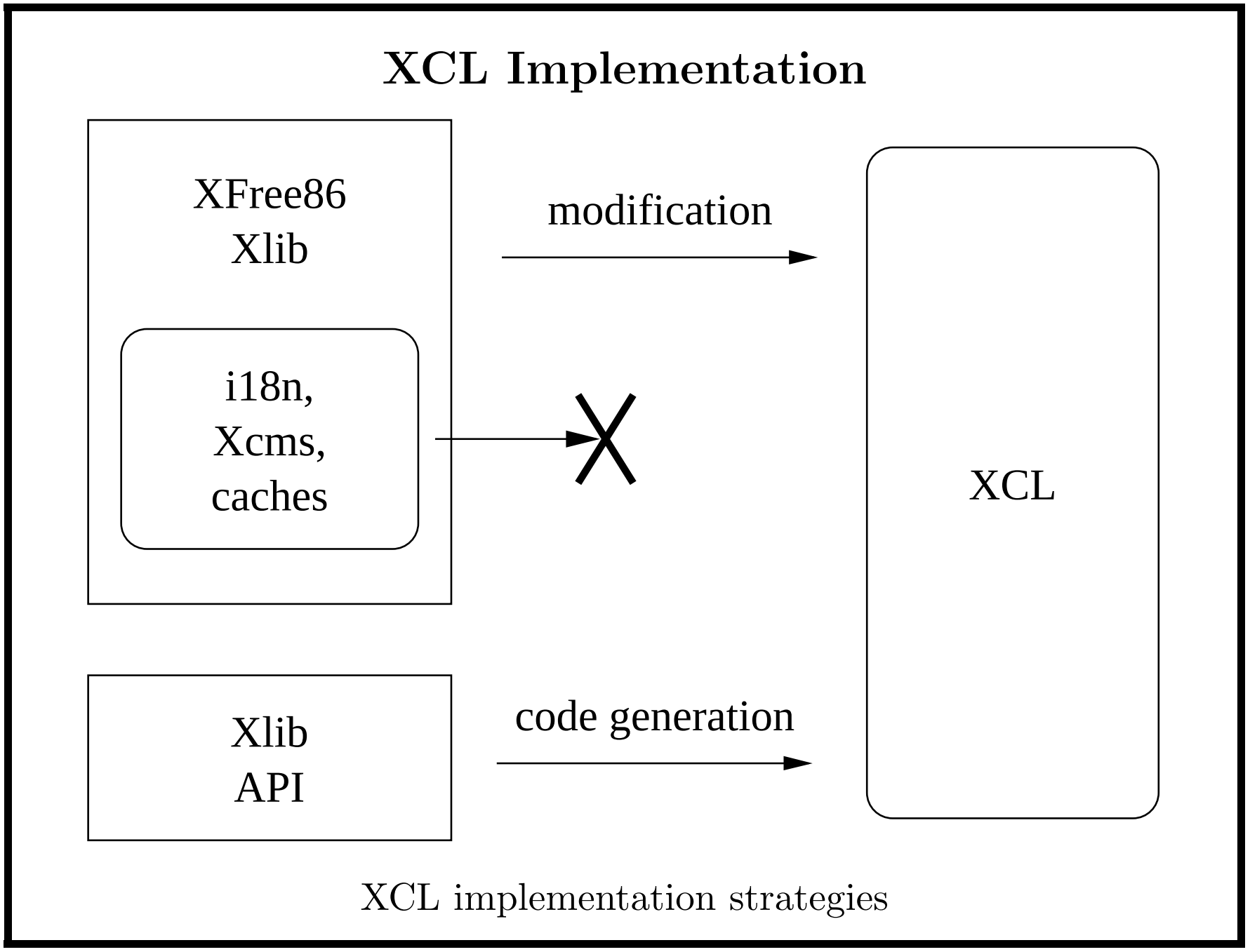


XCL's role compared with Xlib's

Motivations for XCL

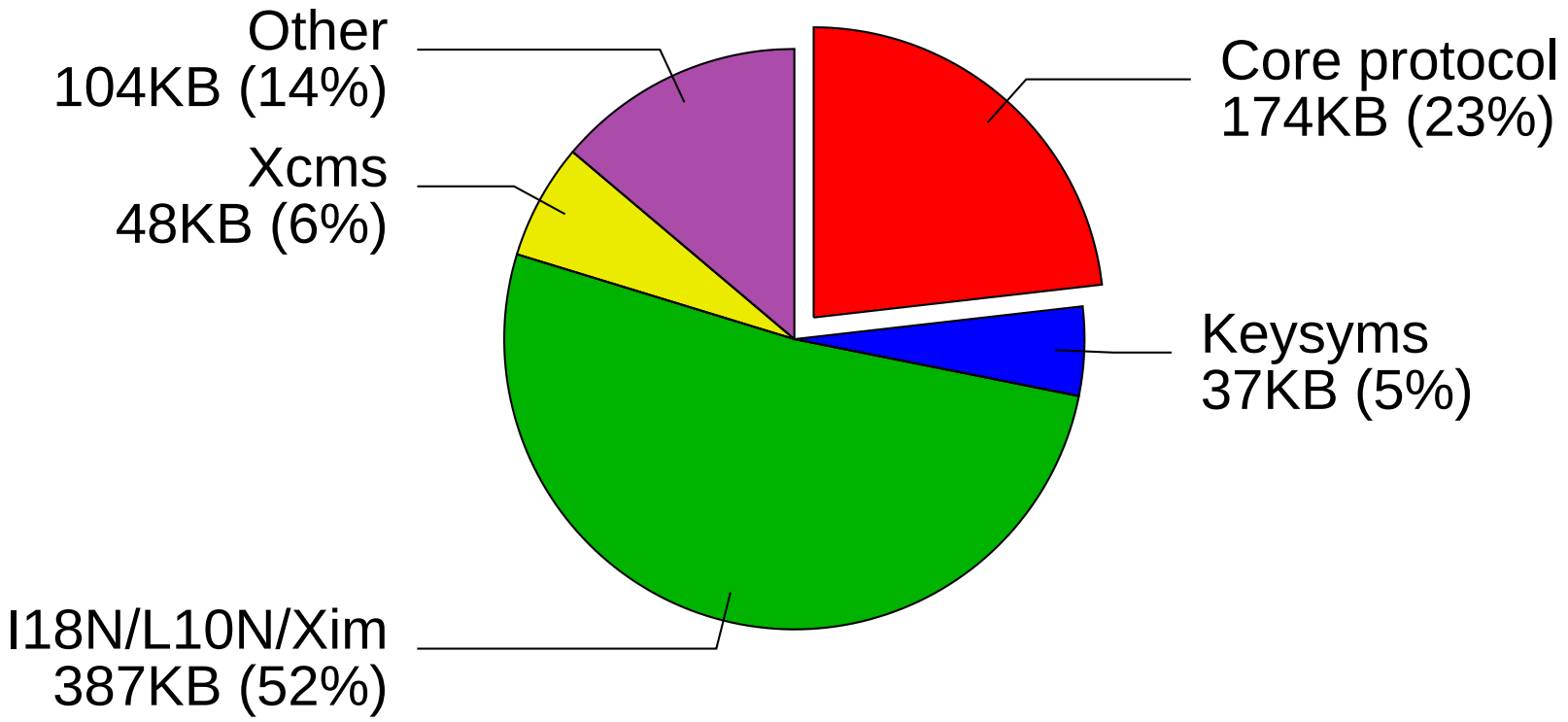
Why re-invent Xlib?

- Significant history: $>$ 15 years worth of software uses Xlib API
- In small environments like hand-held computers, Xlib implementation is big
- XCB implementation is small, but API is incompatible with Xlib applications
- Aid transition to XCB through XCL
- Anticipate possible benefits in performance, latency, reliability



XCL implementation strategies

Xlib Size Analysis



Total: 750KB

XCL Strengths

New design and implementation enables many of XCB's benefits without changing Xlib API.

- Easier maintenance
- More uniform optimization
  - e.g. request marshaling in not just XDrawPoint but also XDrawPoints

XCL Evaluation

- XCL not quite done yet
- Some applications tested without error, e.g.
  - rxvt: perfect behavior
  - gw: nearly perfect
- XCL+XCB is 55KB; Xlib is about 665KB
- XCL performance comparable to Xlib
- Concept has been proven: modern implementation of Xlib API

Future Work

- Complete XCL (extension support)
- Analyze: other Xlib functionality → additional modules
- Validate against existing Xlib-based toolkits (Qt, Gtk+)
- Implement caches on XCB
- Replace core font rendering with Xft

Related Work

- Libraries for other languages:
  - Java client libraries: XTC, Escher
  - Common Lisp: CLX
  - Smalltalk: STIX
  - ML: eXene
- Standalone client: Applegate and Jacobson's 1991 IOCCC entry plays Conway's "Game of Life" on X desktop

Acknowledgements

- The work described in this talk is a joint project with PSU Prof. Bart Massey, XFree86 Core Team member Keith Packard, and high-school student Andy Howe.
- We are greatful for the significant contributions of Jim Gettys, one of the original authors of Xlib and of the design of the X Window System.
- Many thanks to the Computer Science department of Portland State University for supporting this research and enabling us to present the work at the Usenix Annual Technical Conference 2002, and at the XFree86 Technical Conference 2001.

Availability

Current implementations of XCL and XCB are freely available under an MIT-style license at `http://xcb.cs.pdx.edu/`.