Redefining Input in X

Peter Hutterer
peter@cs.unisa.edu.au

why?

multiple input devices!

server device handling

(X Input Extension)

Core events:

Events defined in the core X protocol

XI events:

Events with device ID, defined in the XI protocol extension.

Core pointer

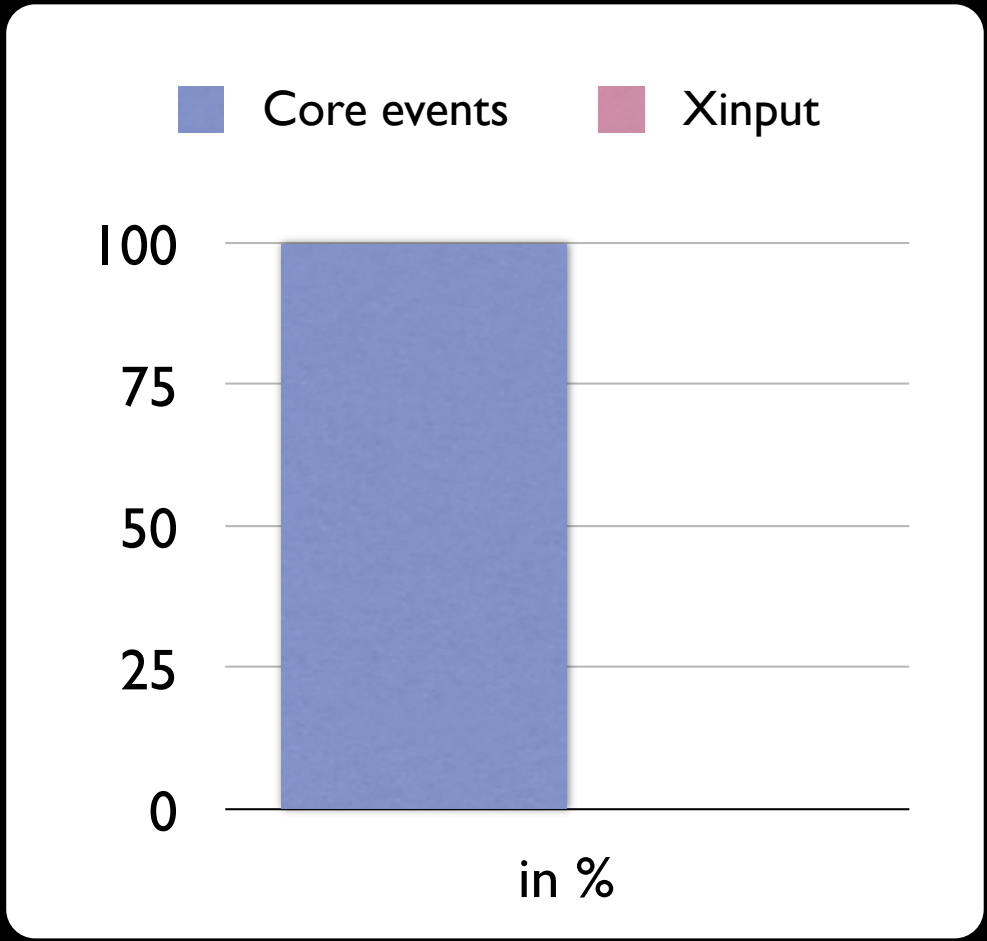
virtual device, cannot send XI events

Core keyboard

virtual device, cannot send XI events

XI devices

no core events but hotpluggy goodness



(numbers rounded to the nearest percent)

one cursor

one keyboard focus

MPX

Multi-Pointer X

Multiple input devices. Anytime. Anywhere.

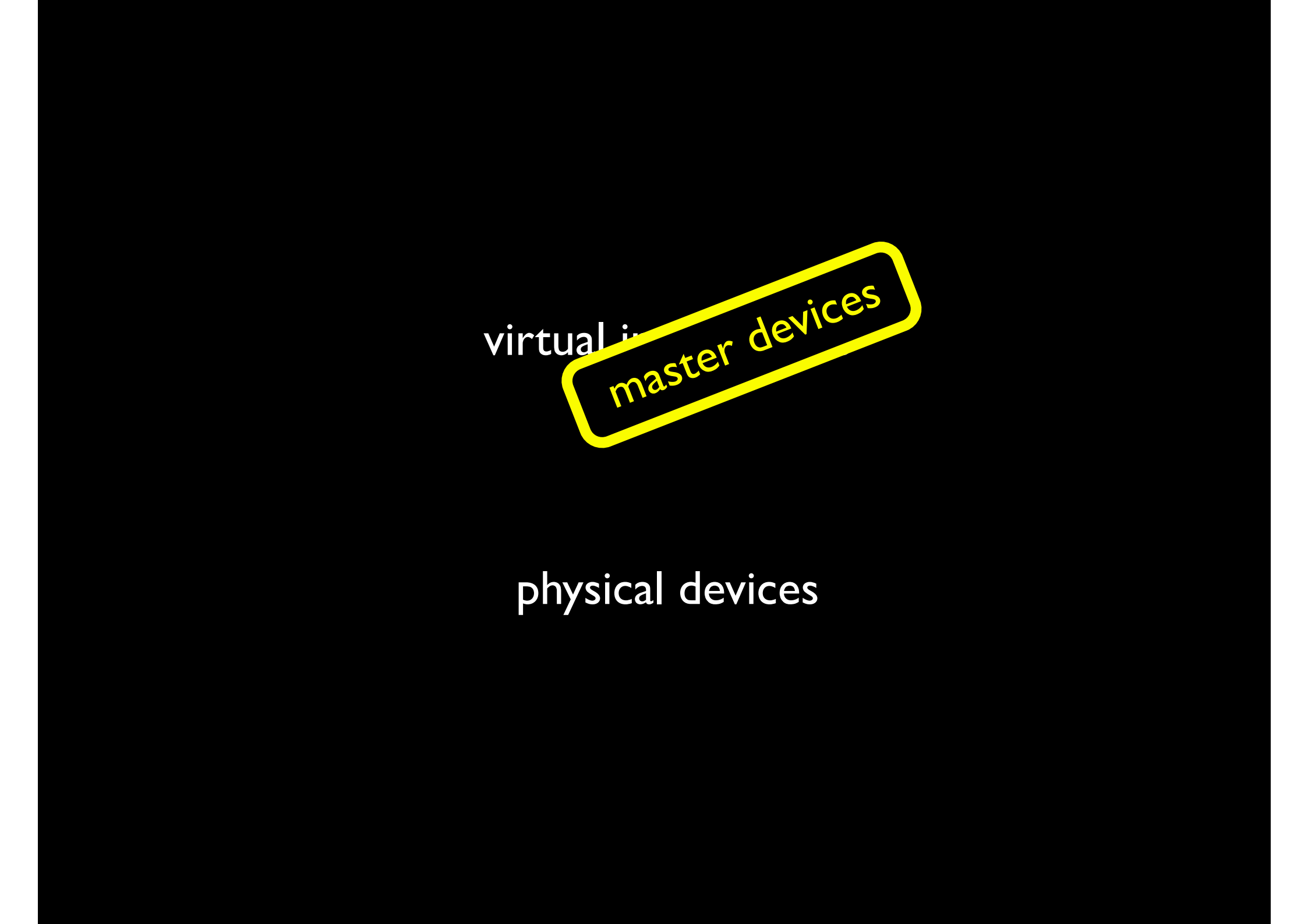
What is an input device?

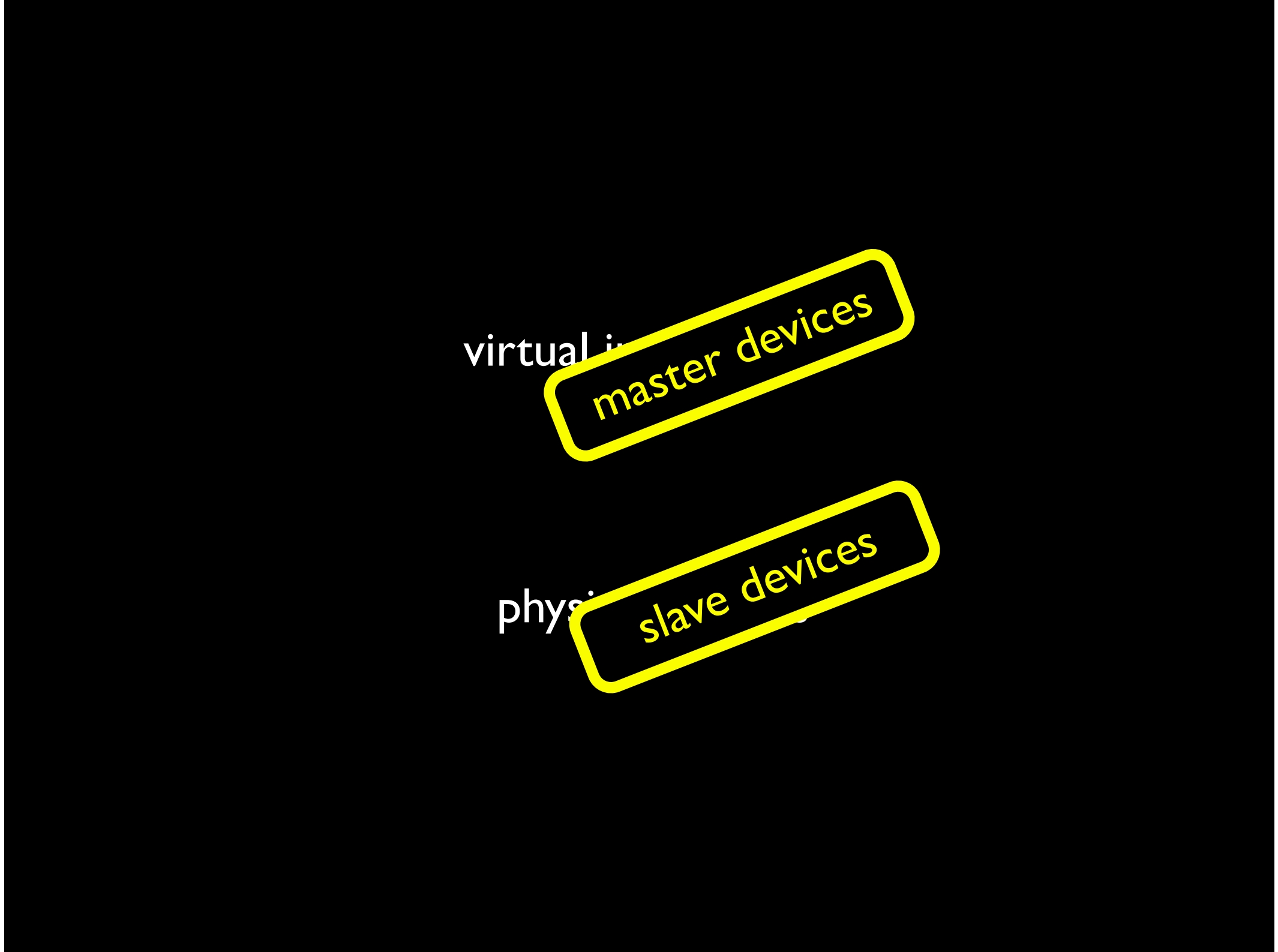
virtual input points

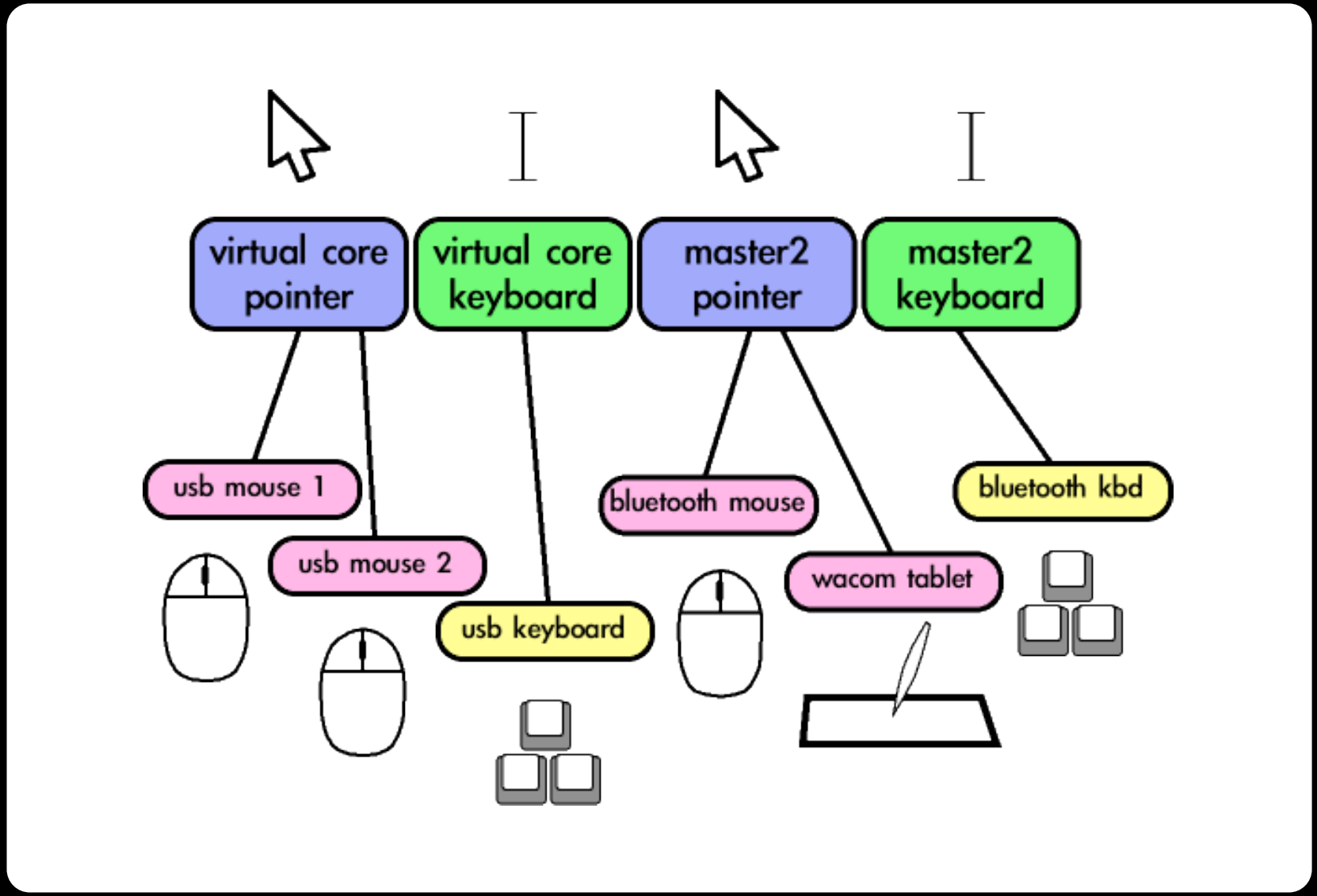
physical devices

virtual input points

physical devices


virtual
master devices
physical devices


virtual
master devices
phys
slave devices


virtual core pointer
virtual core keyboard
master2 pointer
master2 keyboard
usb mouse 1
usb mouse 2
usb keyboard
bluetooth mouse
wacom tablet
bluetooth kbd

Master pointer are cursors

Master keyboard are keyboard foci

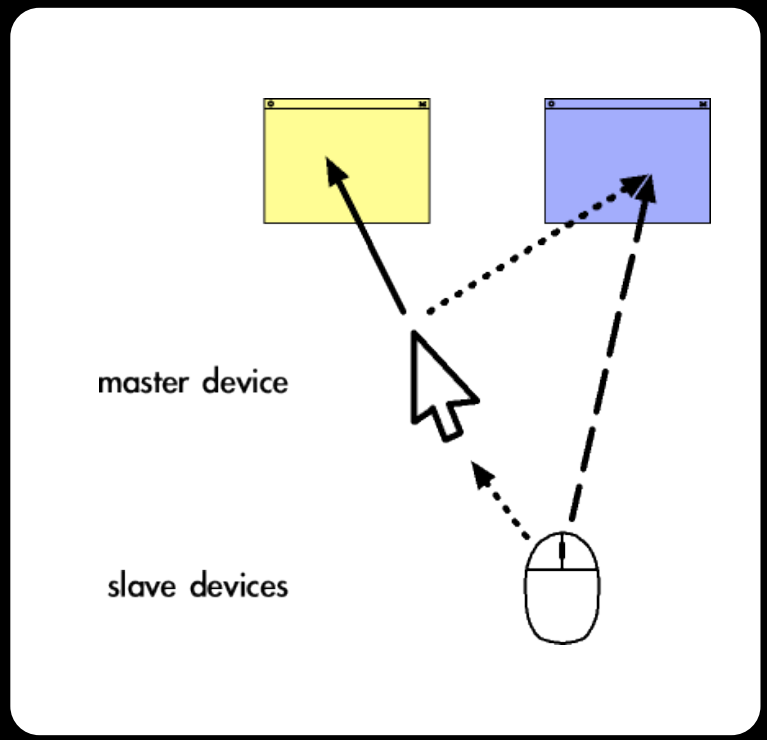
- can send core events
- are always virtual

Slave devices *may* be attached to a master.

Slave devices route event through master.

3 events per event:

- XI event from slave
- core event from master
- XI event from master



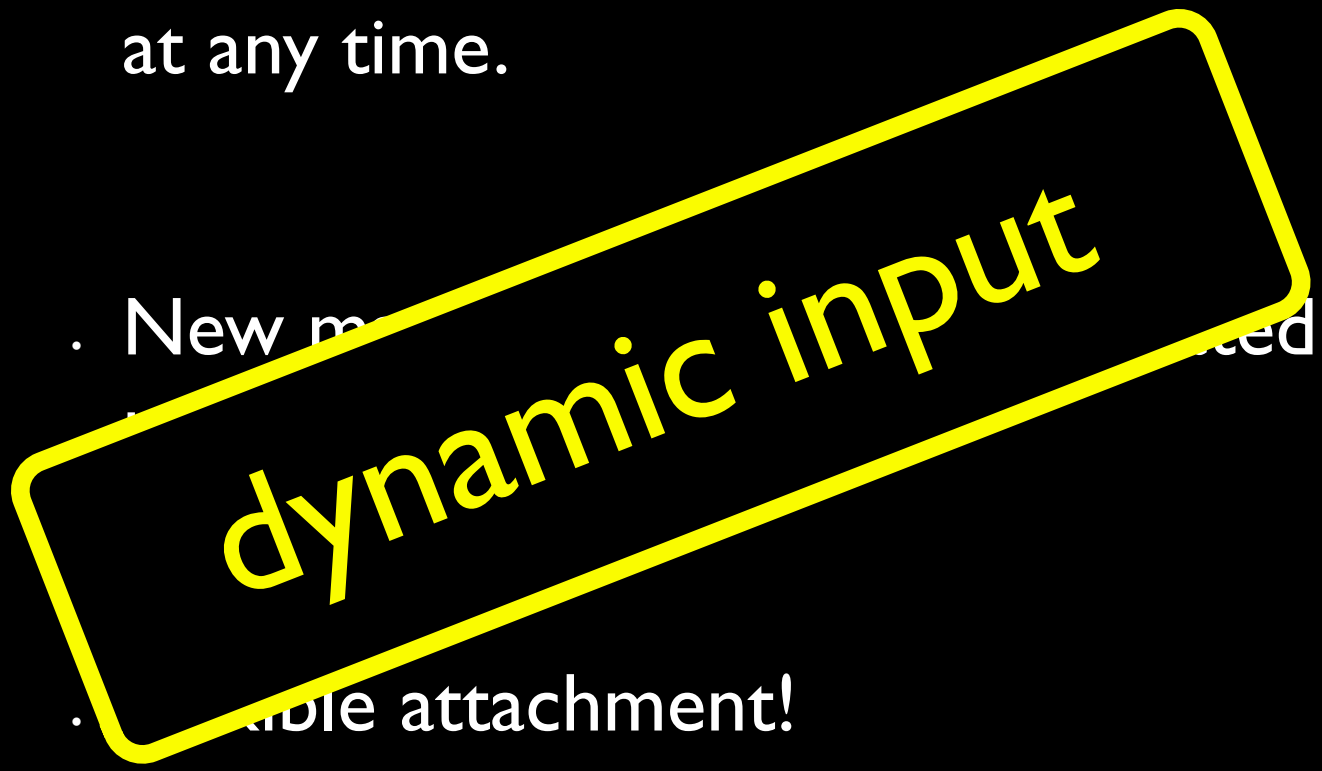
- Slave devices can be plugged and unplugged at any time.
- New master devices can be created and removed at any time.
- Flexible attachment!

- Slave devices can be plugged and unplugged at any time.
- New master devices can be created and removed at any time.
- Flexible attachment!

- Slave devices can be plugged and unplugged at any time.

- New m ted and

- ible attachment!



Plan A:

- Applications can program multi-user interfaces with the XInput API
  - per-device events
  - per-device APIs
  - eternal happiness

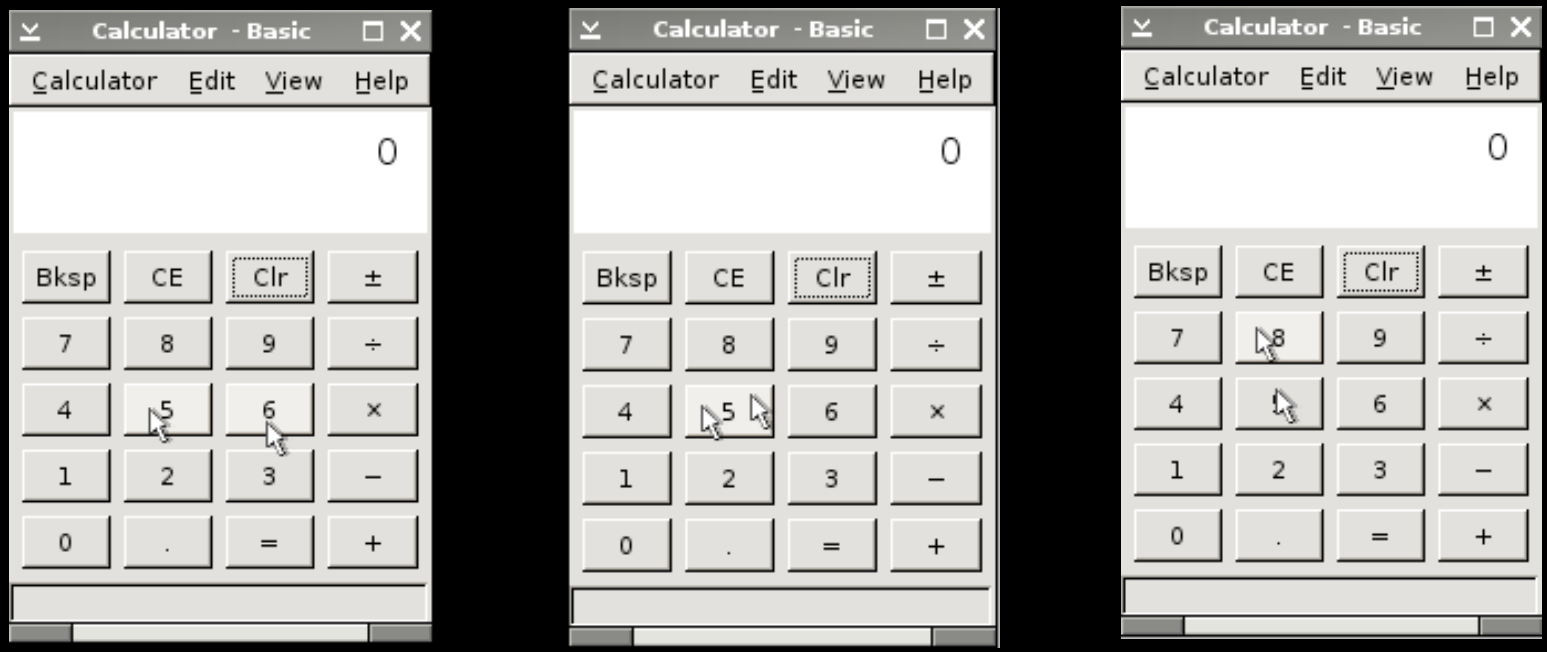
What about the other 100%?

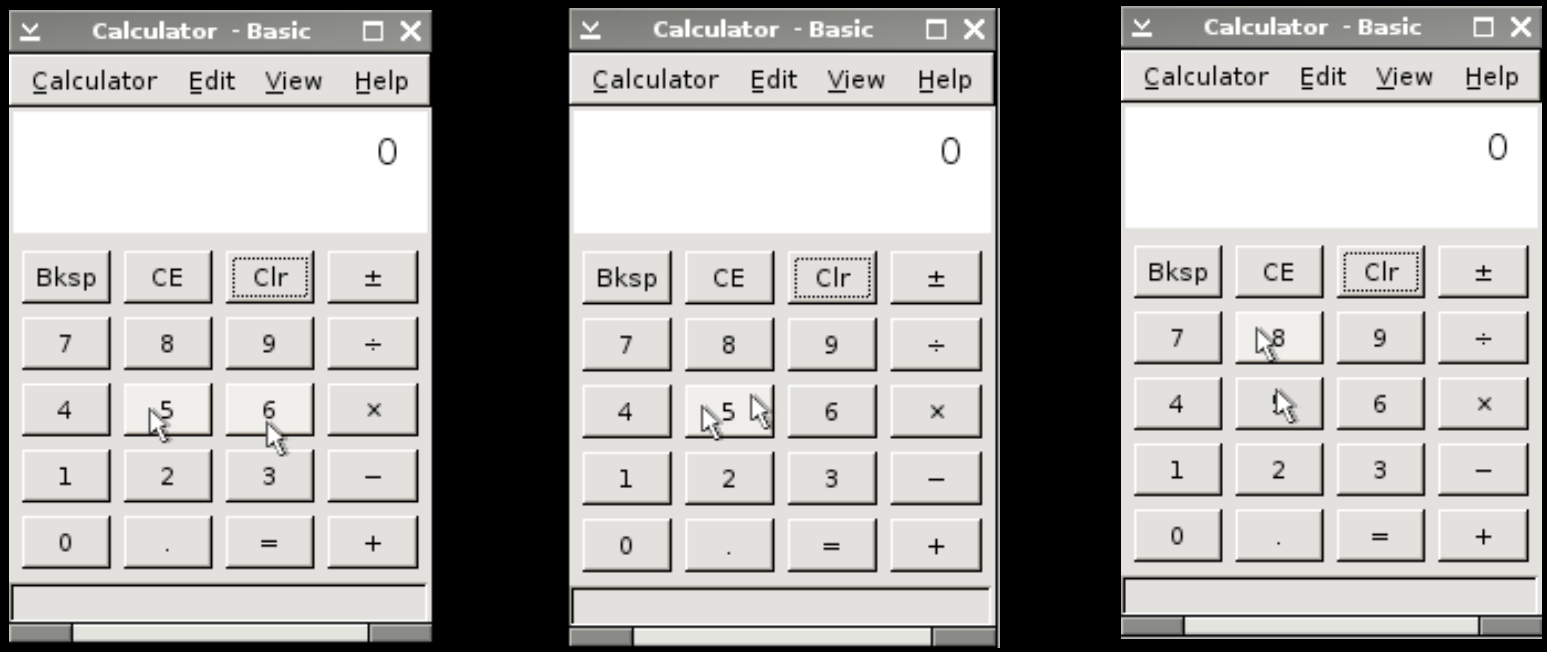
Multiple devices in standard apps

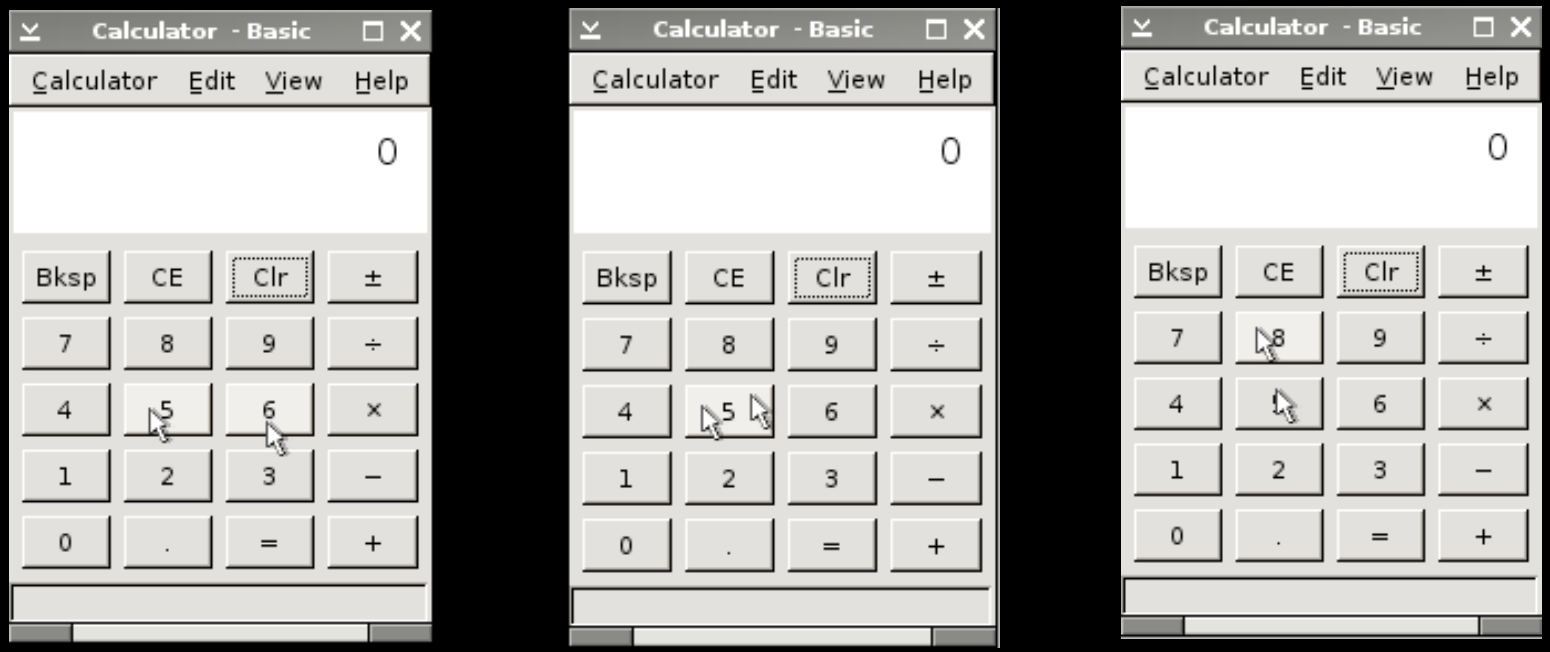
applied insanity 101

The three capital offences:

- Duplicate enter/leave events
- Inconsistent event sequences.
- Which device was it again?


Calculator - Basic
Calculator Edit View Help
0
Bksp CE Clr ±
7 8 9 ÷
4 5 6 ×
1 2 3 −
0 . = +
Calculator - Basic
Calculator Edit View Help
0
Bksp CE Clr ±
7 8 9 ÷
4 5 6 ×
1 2 3 −
0 . = +
Calculator - Basic
Calculator Edit View Help
0
Bksp CE Clr ±
7 8 9 ÷
4 6 ×
1 2 3 −
0 . = +

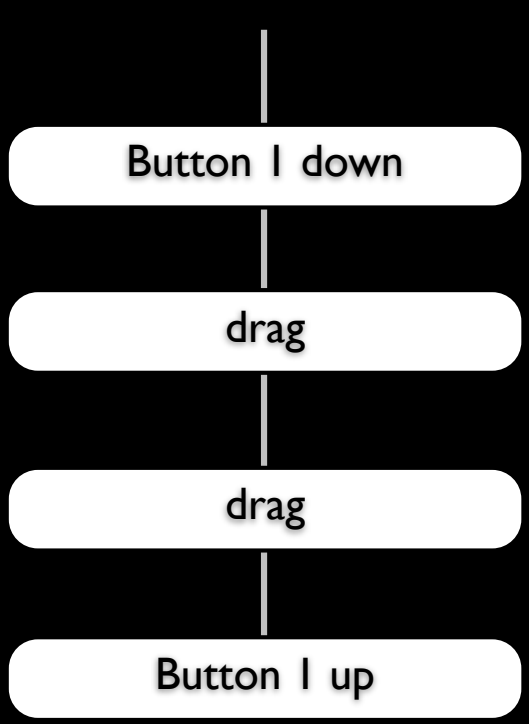

Calculator - Basic
Calculator Edit View Help
0
Bksp CE Clr ±
7 8 9 ÷
4 5 6 ×
1 2 3 −
0 . = +
Calculator - Basic
Calculator Edit View Help
0
Bksp CE Clr ±
7 8 9 ÷
4 5 6 ×
1 2 3 −
0 . = +
Calculator - Basic
Calculator Edit View Help
0
Bksp CE Clr ±
7 8 9 ÷
4 6 ×
1 2 3 −
0 . = +

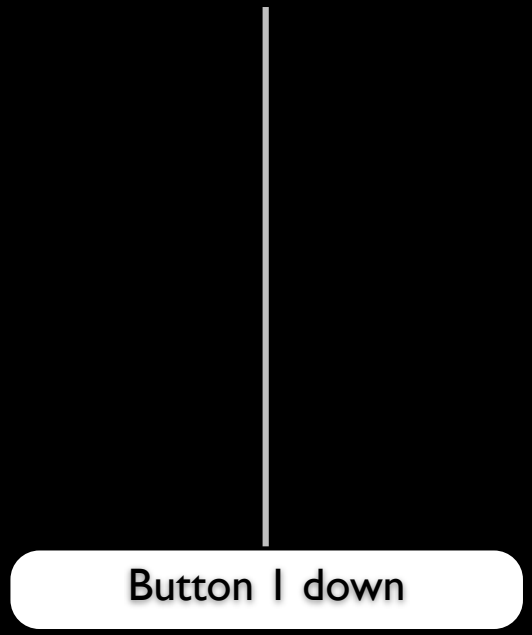


Solution:

One Enter/Leave event per window.

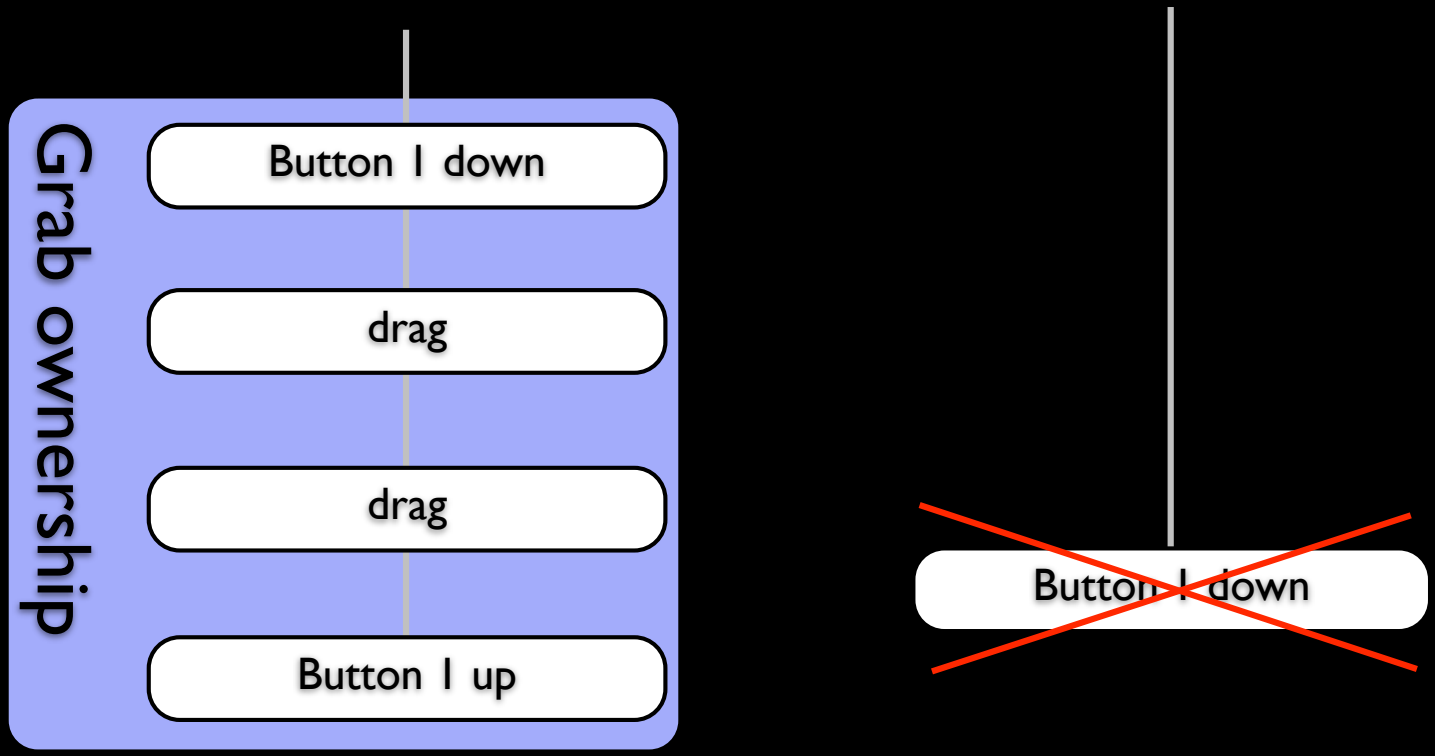
Inconsistent event sequences:



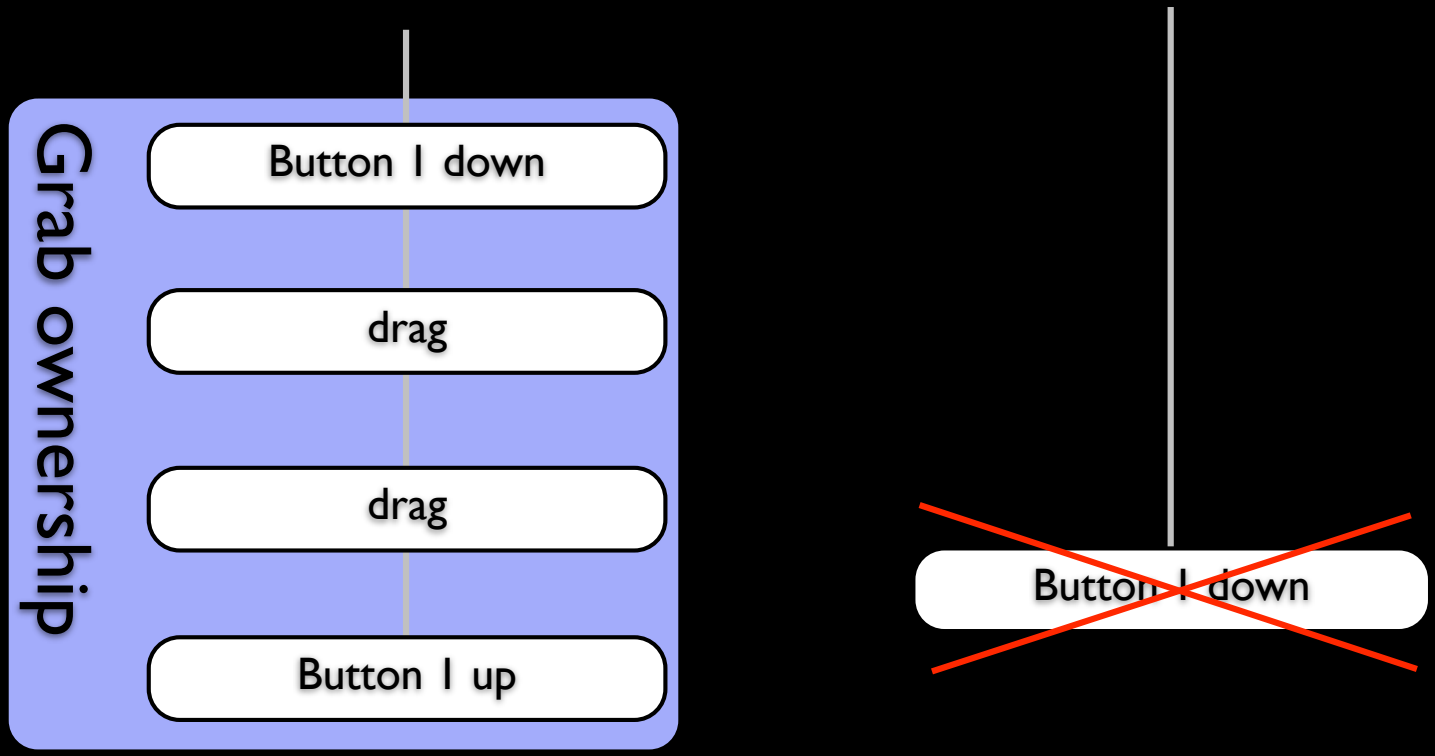


Panic!

Inconsistent event sequences:



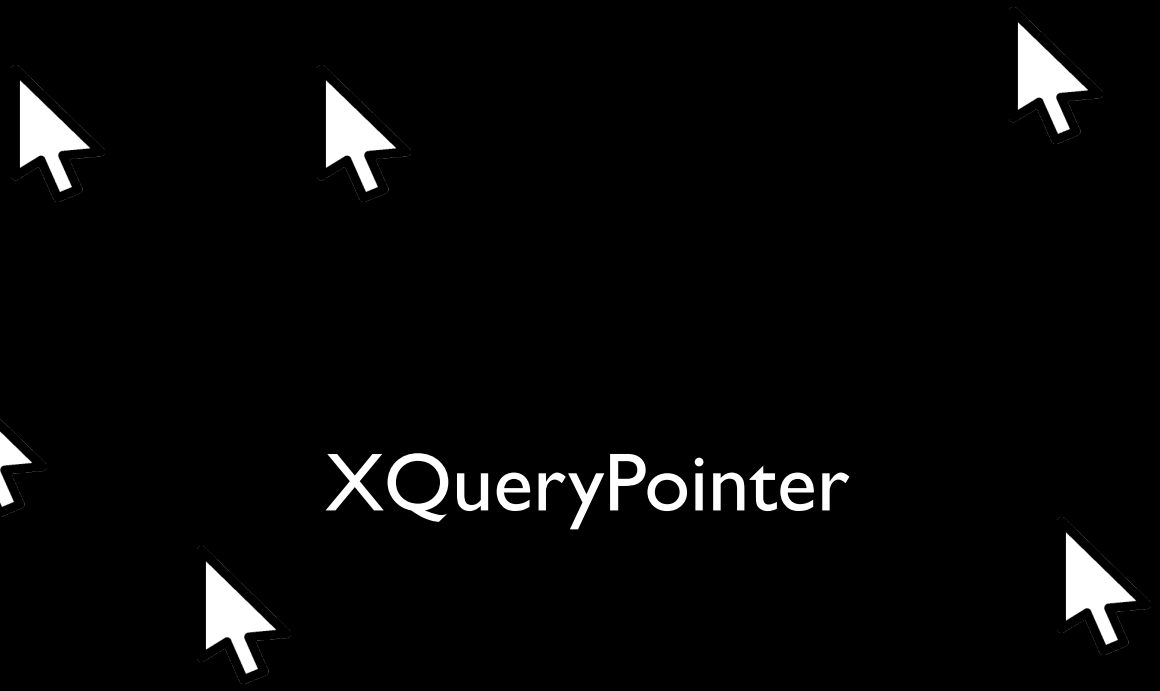
Inconsistent event sequences:



Solution:

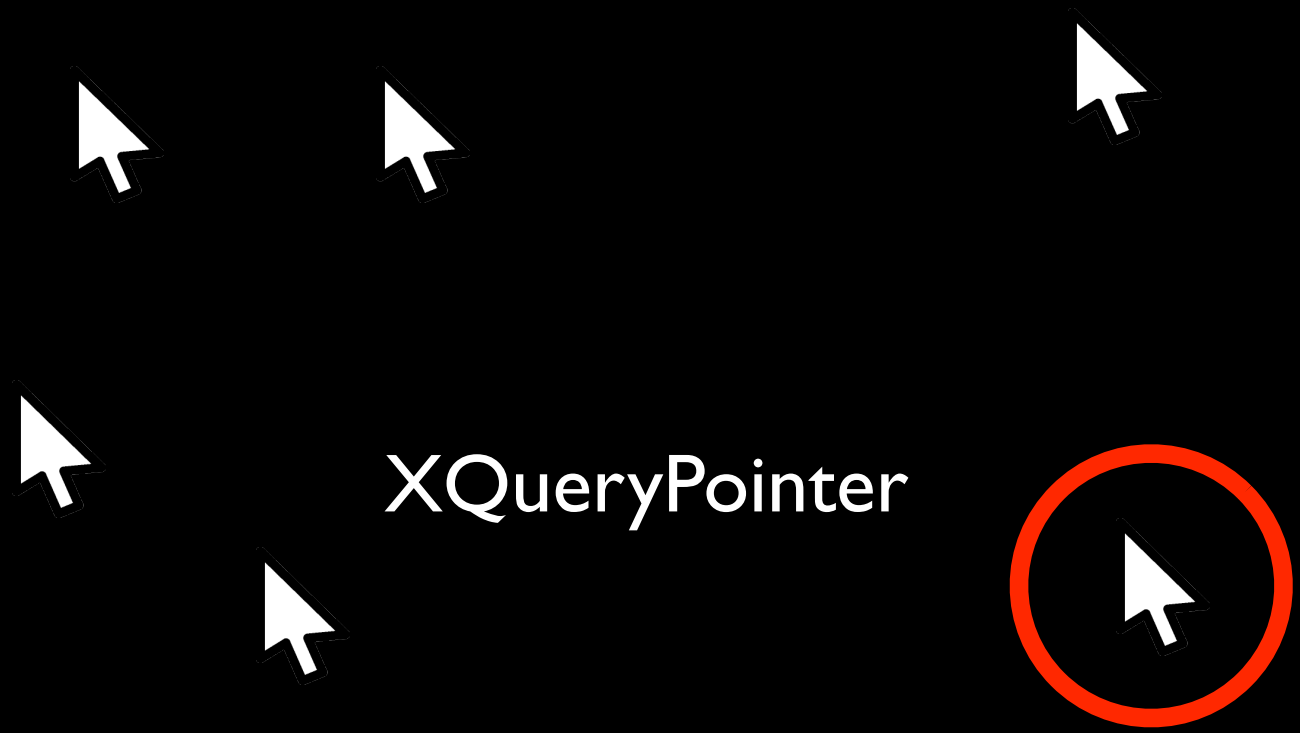
Only one device can send events during a core grab.

Ambiguity



Ambiguity

Ambiguity



Solution:

Each client has a ClientPointer

Plan B:

- Support multiple devices in standard apps:
  - Adjust the event stream
  - Grab ownership
  - ClientPointer
  - fix all the other little things that come up

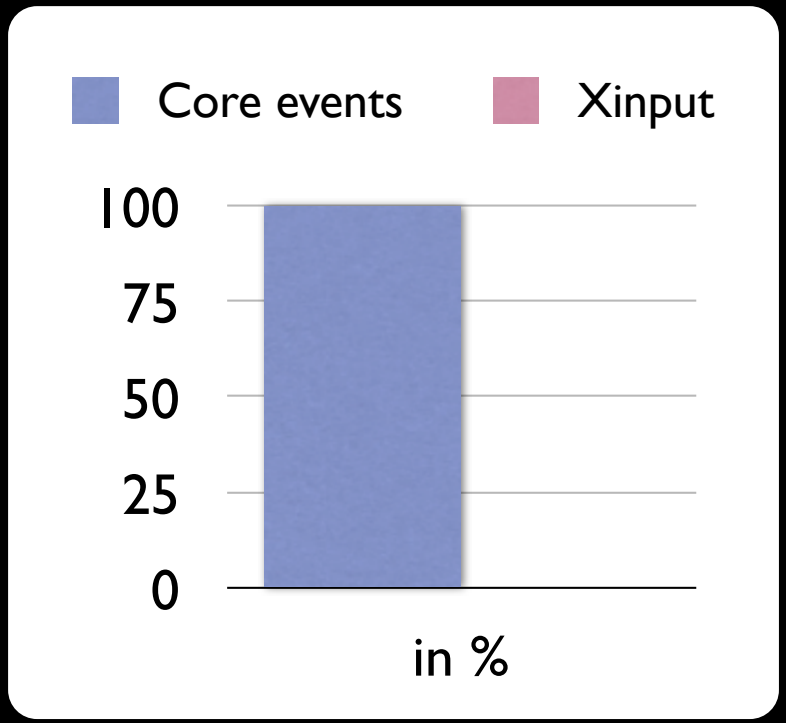
rinse. wash. repeat.

We can use standard apps.

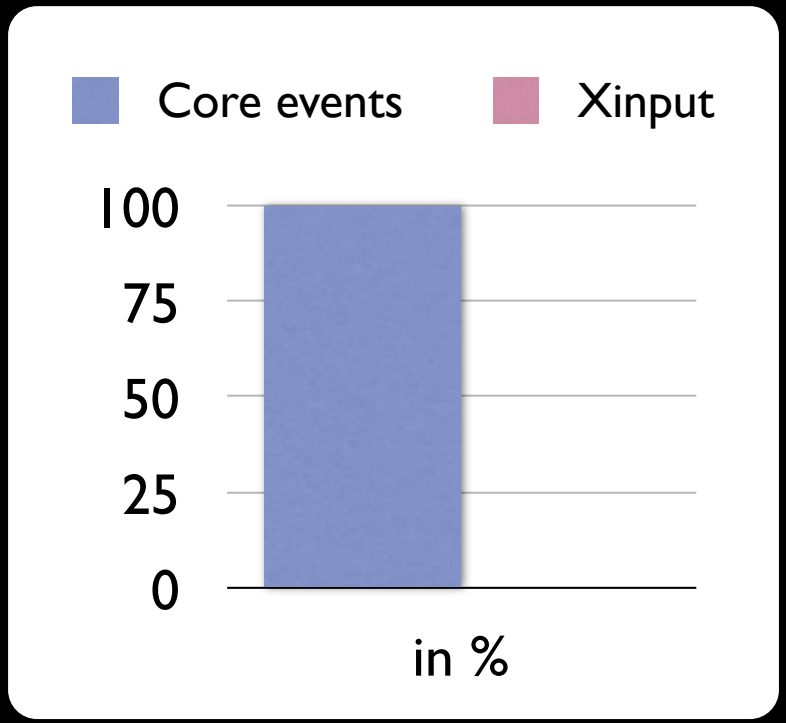
We can write new apps.

We haven’t broken anything!

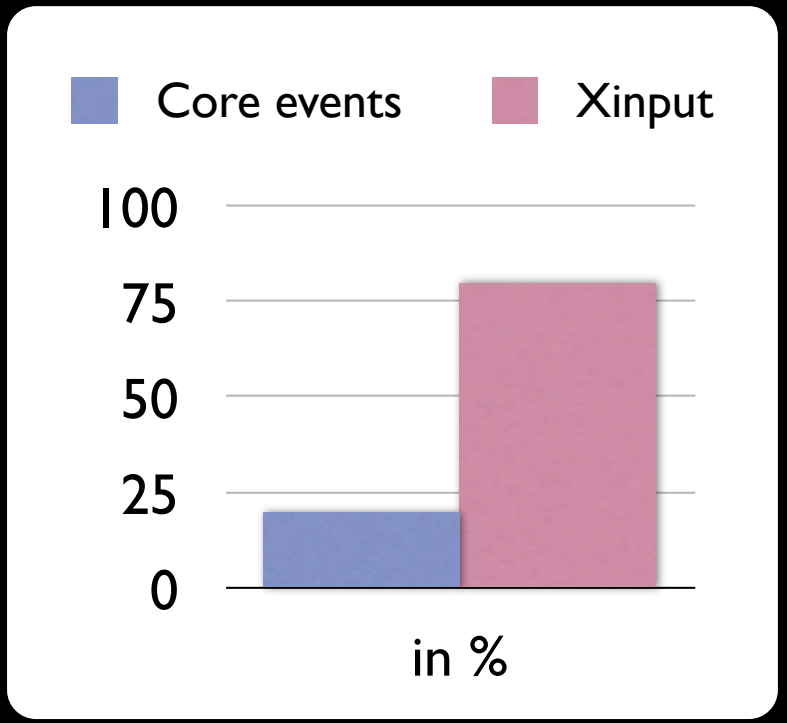
The Future?



2008


Core events
Xinput
100
75
50
25
0
in %

2008


Core events
Xinput
100
75
50
25
0
in %

2008 + n

Support device hierarchy and hotplugging.

easy

Allow simultaneous interaction **at all times**.

hard

Adjust to group-work and two-handed input.

insane

Master devices may switch state when a slave device sends an event.

- Absolute to relative
- Different number of axes
- Change in resolution
- Number of buttons change
- Different capabilities
- etc.

Window managers

How can we avoid optical occlusion?

How can we do smart window placement for multiple users?

How can we even know who owns which devices?

Applications

Good bye menus.

Good bye traditional widgets.

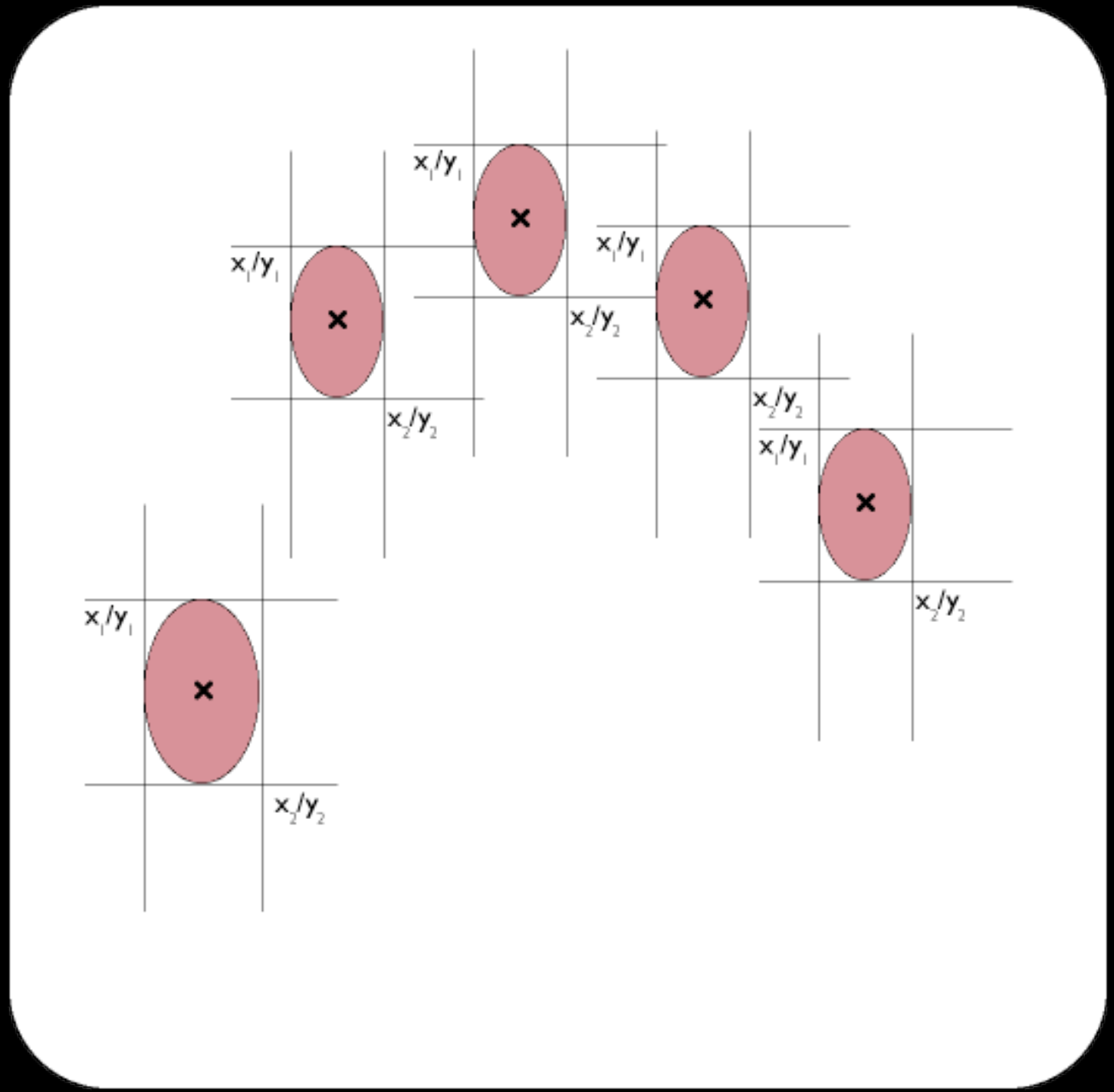
Good bye WIMP?

demo

blob events

touch devices

torch devices


x1/y1
x2/y2
x1/y1
x2/y2
x1/y1
x2/y2
x1/y1
x2/y2

Multiple Area Input Device (MAID)

X can only do point-based devices.

A new event can fix it:

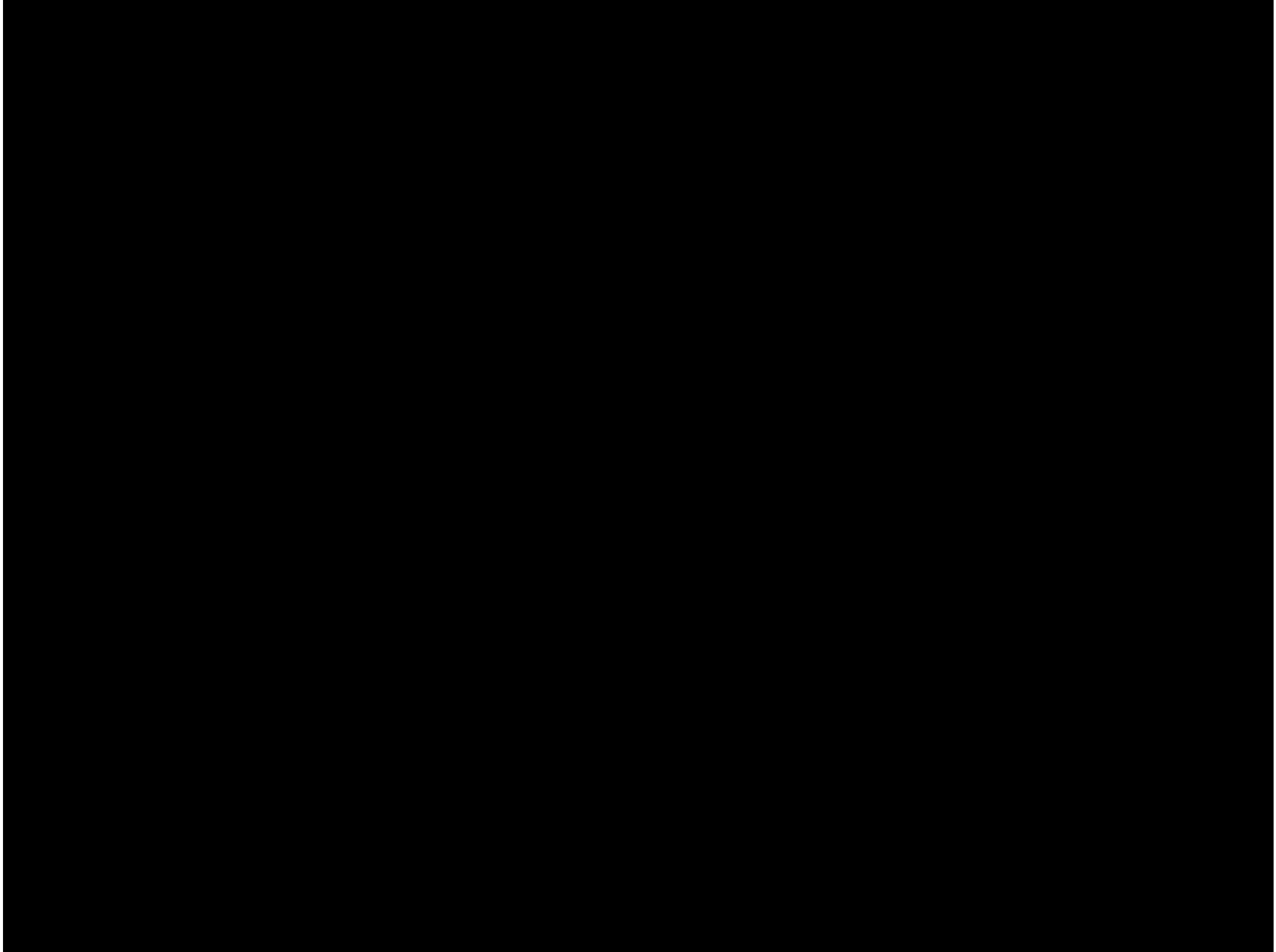
- XBlobEvent
  - hotspots (automatic pointer emulation)
  - areas + data (high-resolution data from devices)
  - elevation (hover effects)
  - rotation
  - intensity (pressure, light intensity)
  - identifiers (through X Atoms)
  - buttons

Why BlobEvents?

- applications are device independent
- hot-plug capable
  - multiple touch screens simultaneously
- query-able
  - adjust UI to type of touchscreen
- automatic pointer emulation
  - use with legacy applications

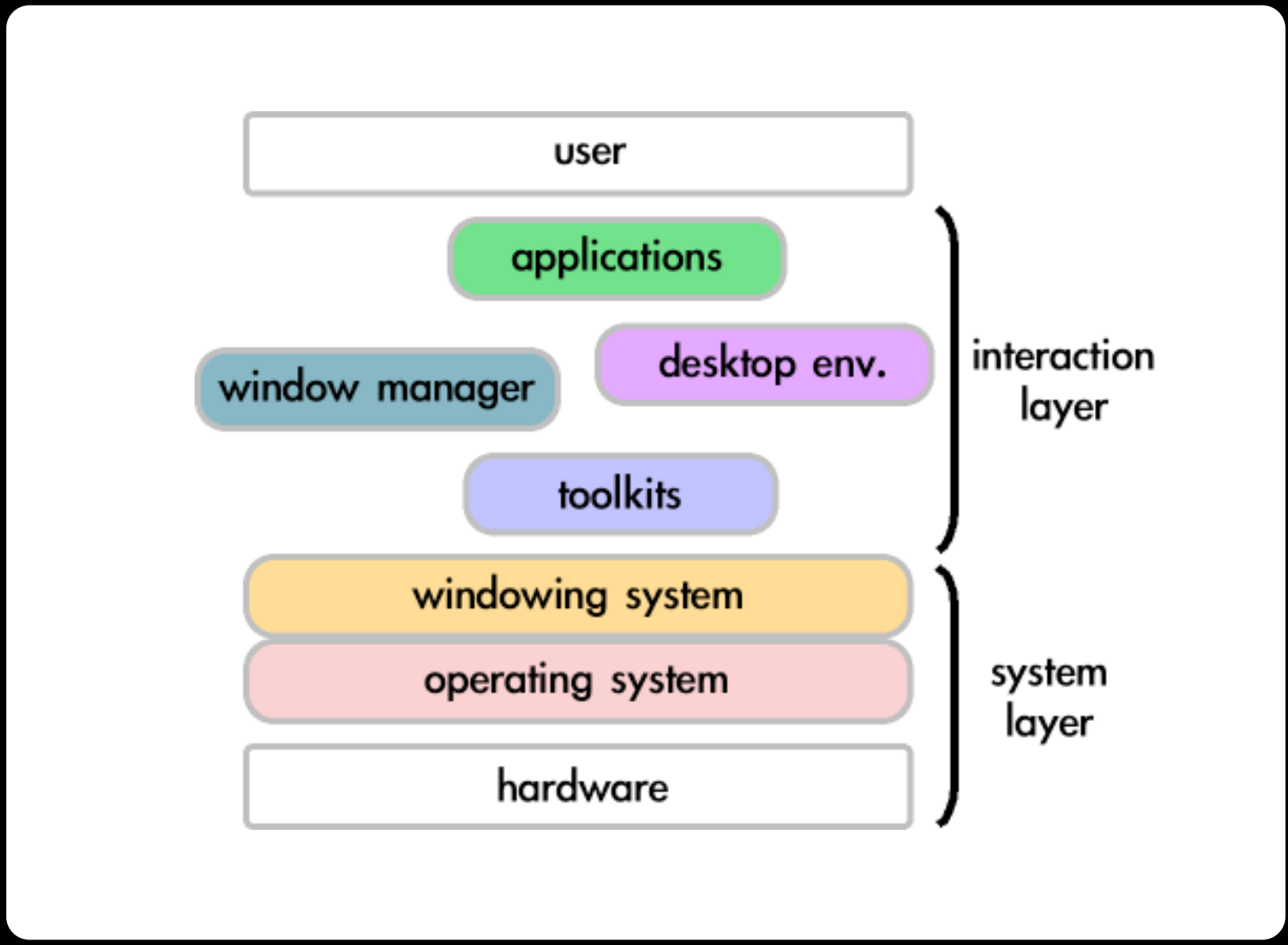
but

How do we deal with transient devices?



Devices. Lots of them.

Just one piece of the puzzle


user
applications
desktop env.
window manager
toolkits
interaction layer
windowing system
operating system
hardware
system layer

How will we use it?

MPX

Peter Hutterer

peter@cs.unisa.edu.au

http://wearables.unisa.edu.au/mpx/