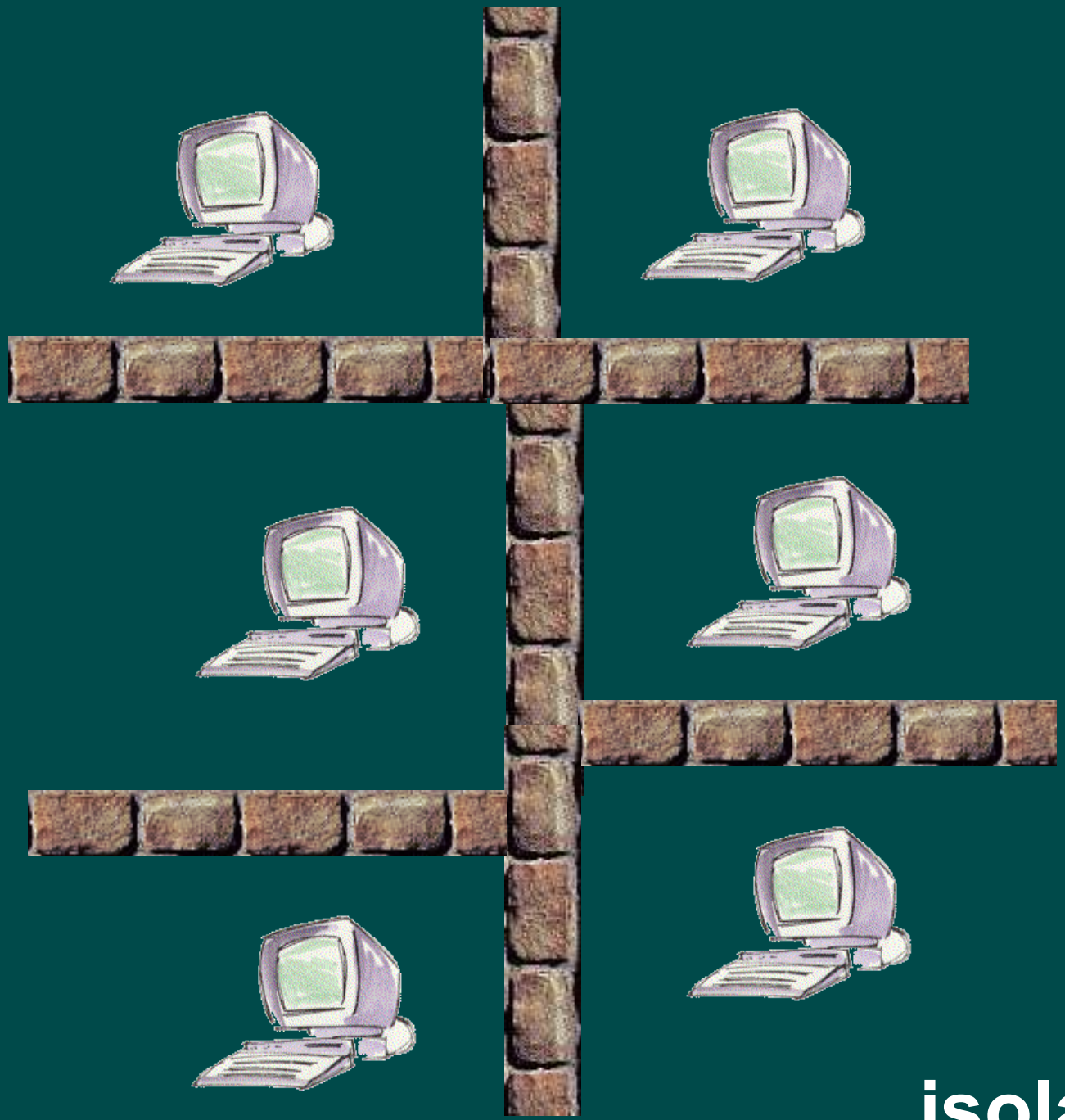
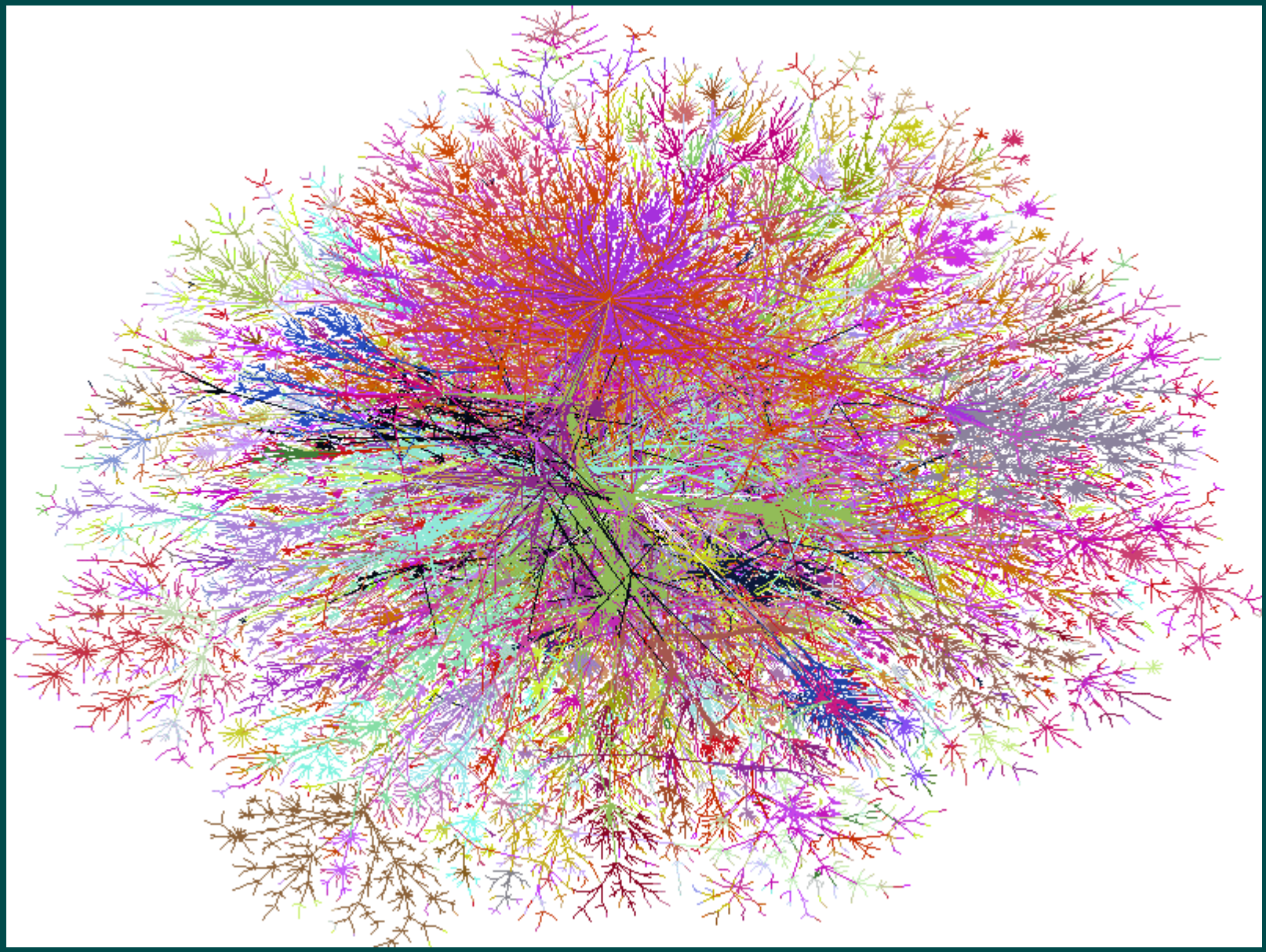
THINC: A Virtual Display Architecture for Thin-Client Computing

Ricardo A. Baratto, Leonard N. Kim, Jason Nieh
Network Computing Laboratory
Columbia University



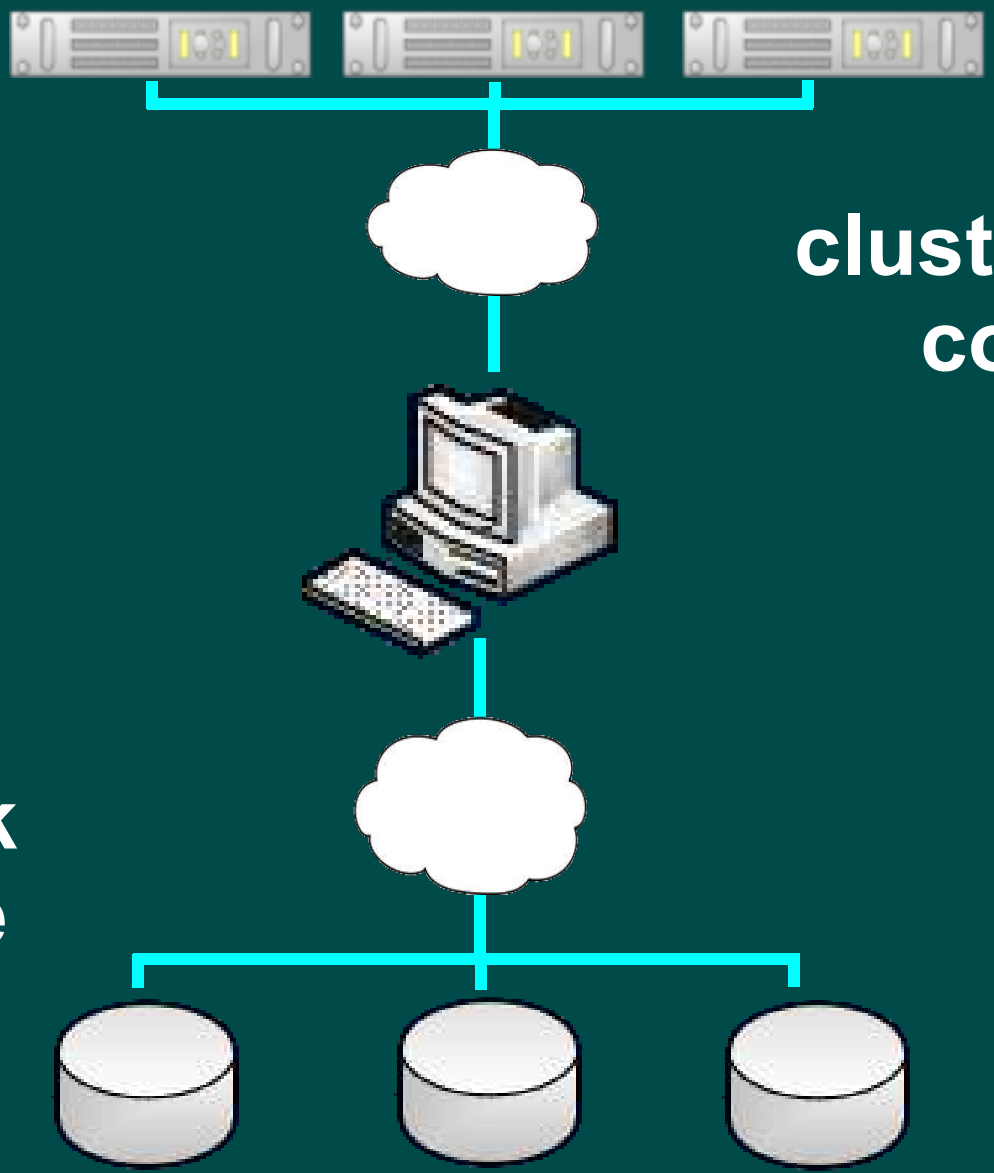
isolation...

...connectivity



Source: Internet Mapping Project (http://research.lumeta.com/ches/map/)

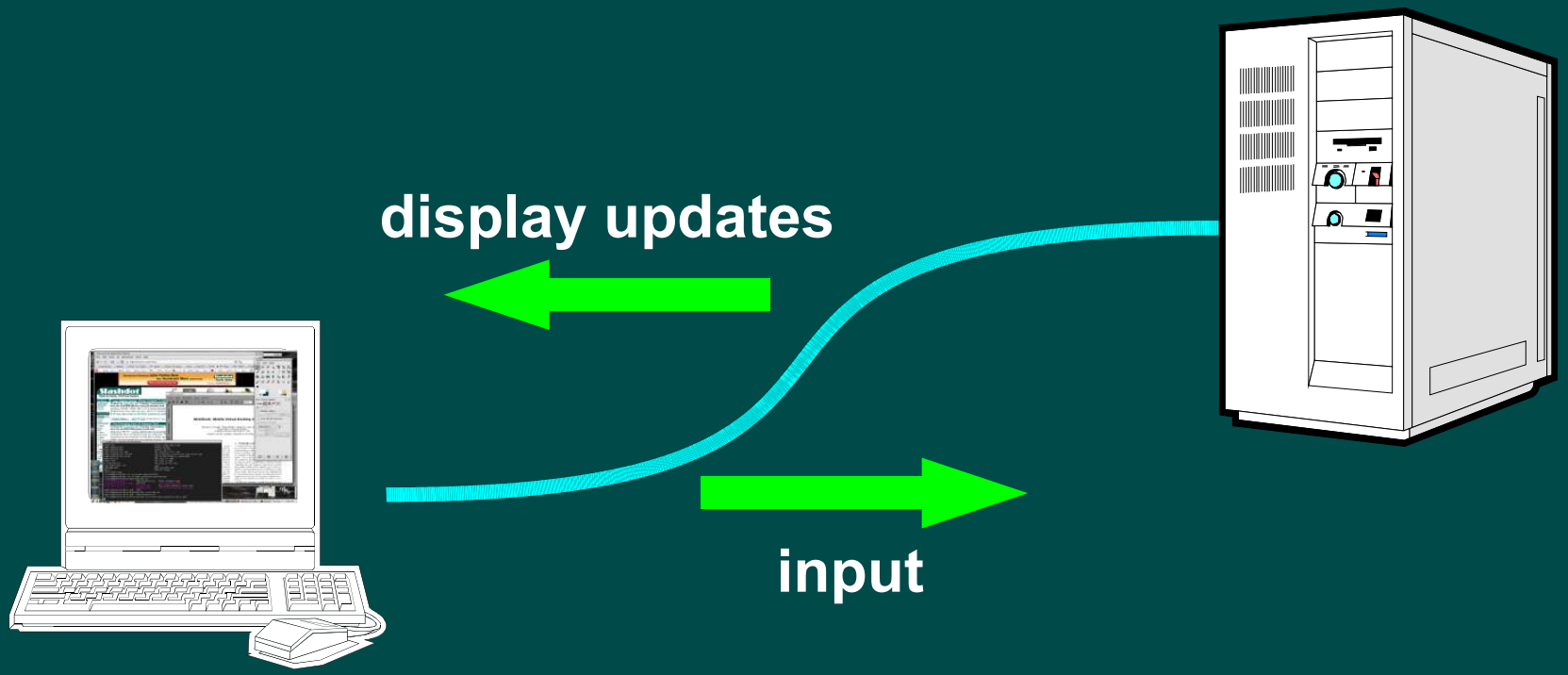
dis-integration of the computer



clusters and grid computing

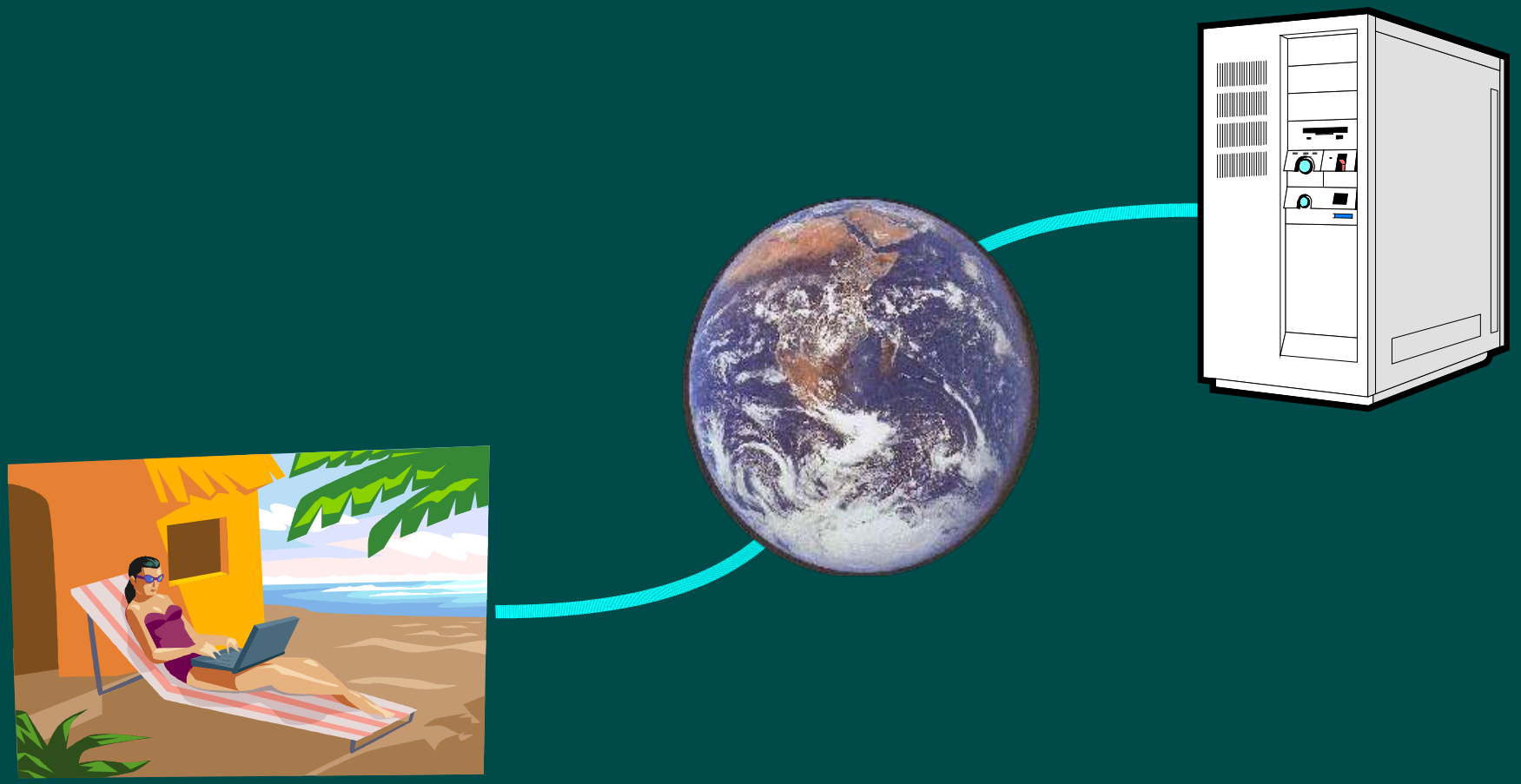
network storage

remote display



benefits

ubiquitous access



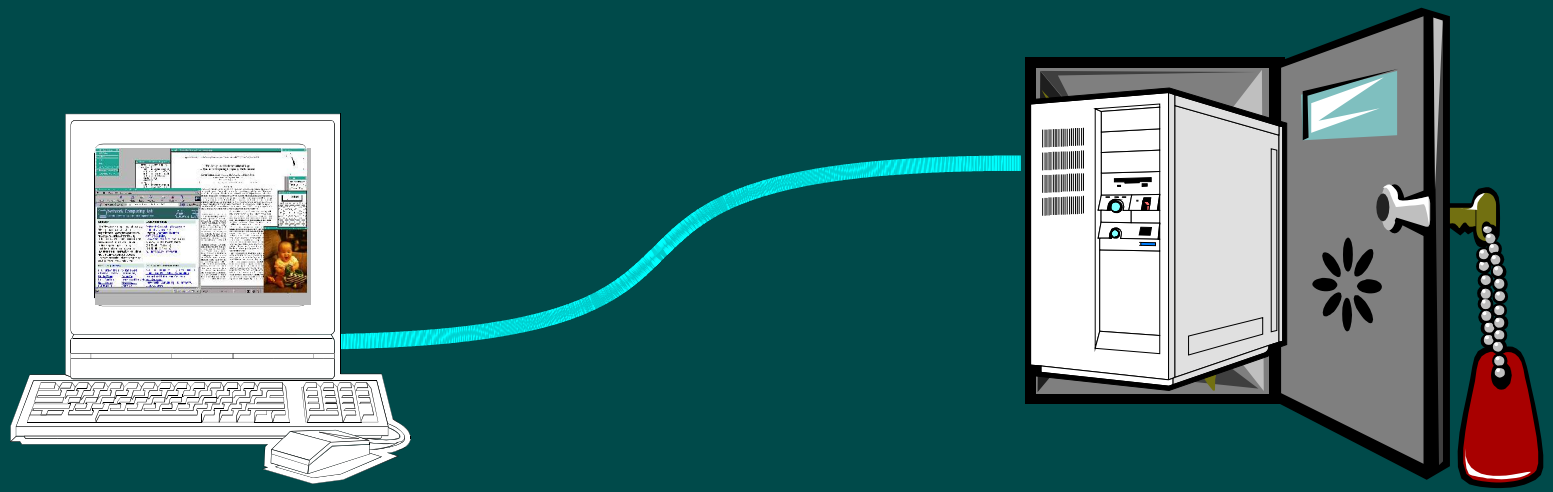
remote collaboration

online help


www.zeebarf.com
zeebarf00
"Okay your father
managed to get a mouse.
Now how do we use it?"

thin clients

application processing and data



stateless client

secure server room

existing systems

CITRIX®

REAL VNC

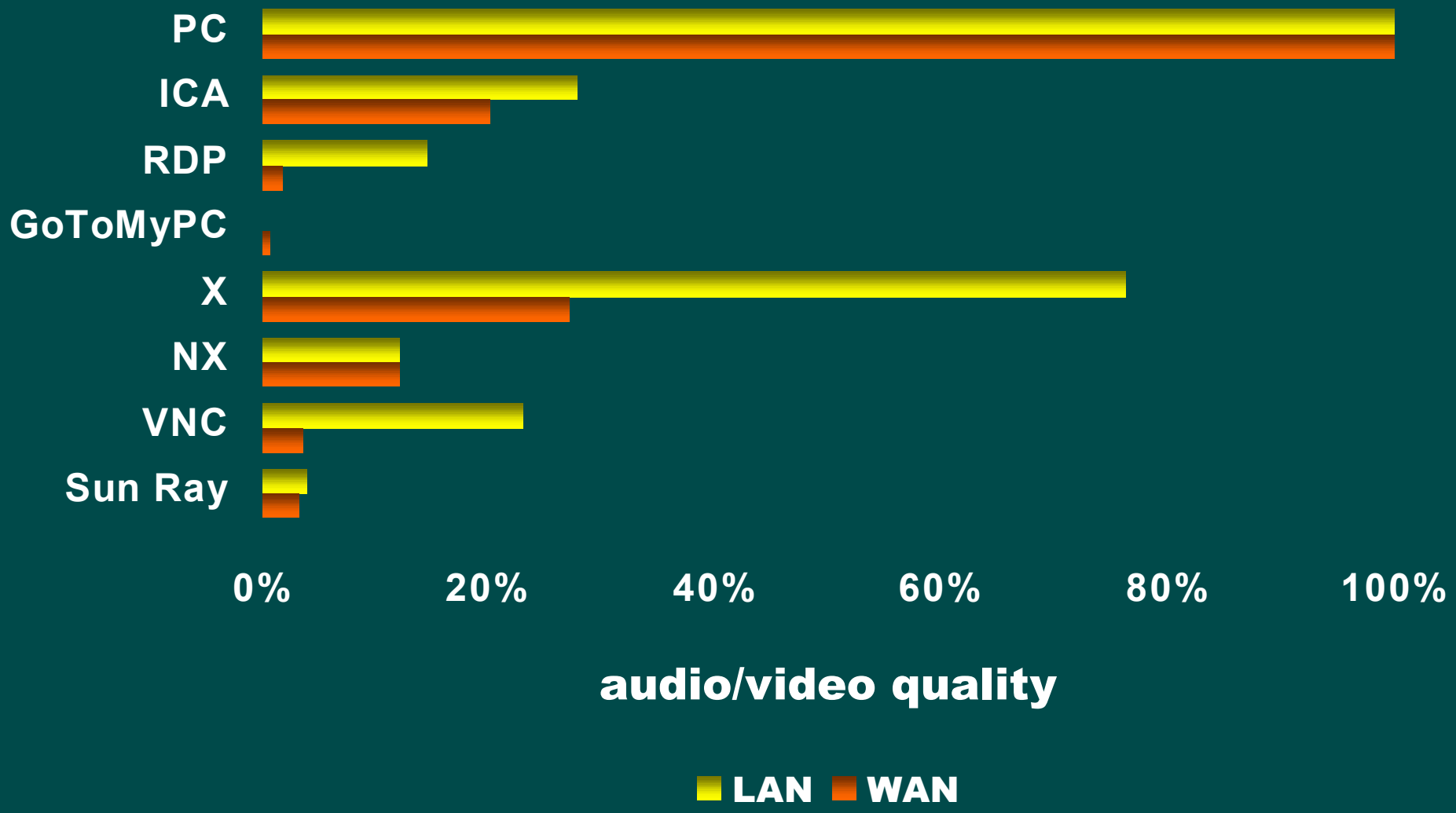
Microsoft Windows xp

GoToMyPC®

NOMACHINE

BUILDING THE NETWORK COMPUTING ON THE POWER OF X

existing performance problem



THINC

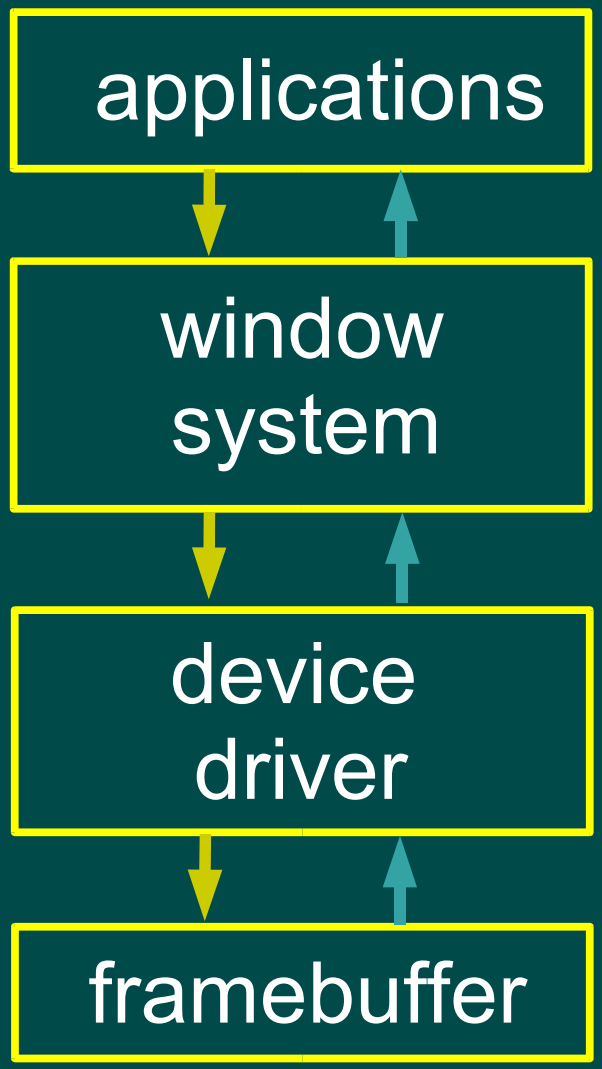
virtual display architecture

high performance remote display

transparent operation

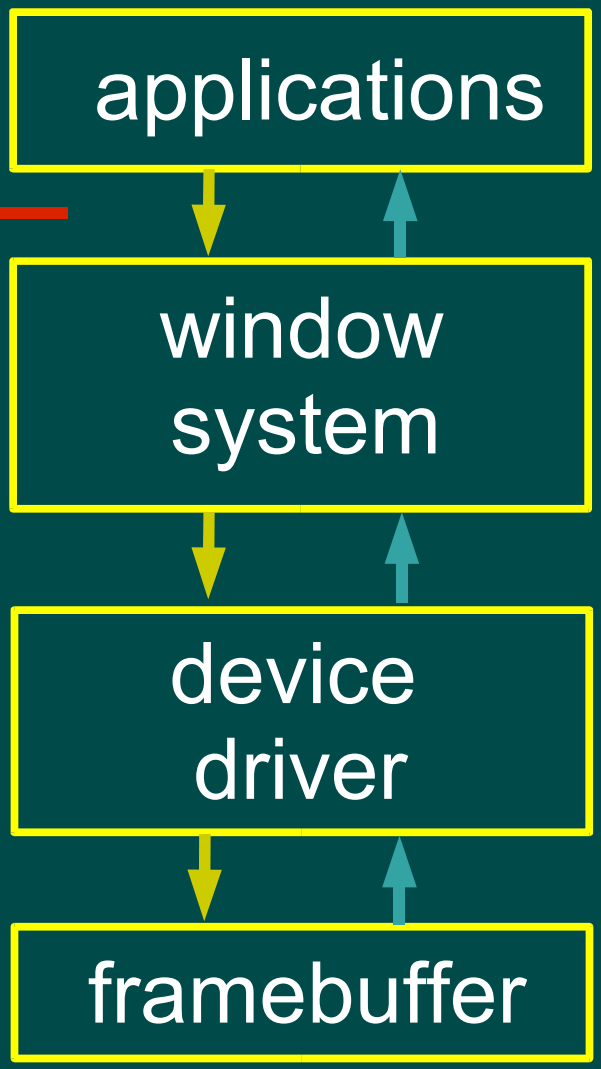
- system architecture
- display protocol
- translation
- delivery

system architecture


applications
window system
device driver
framebuffer

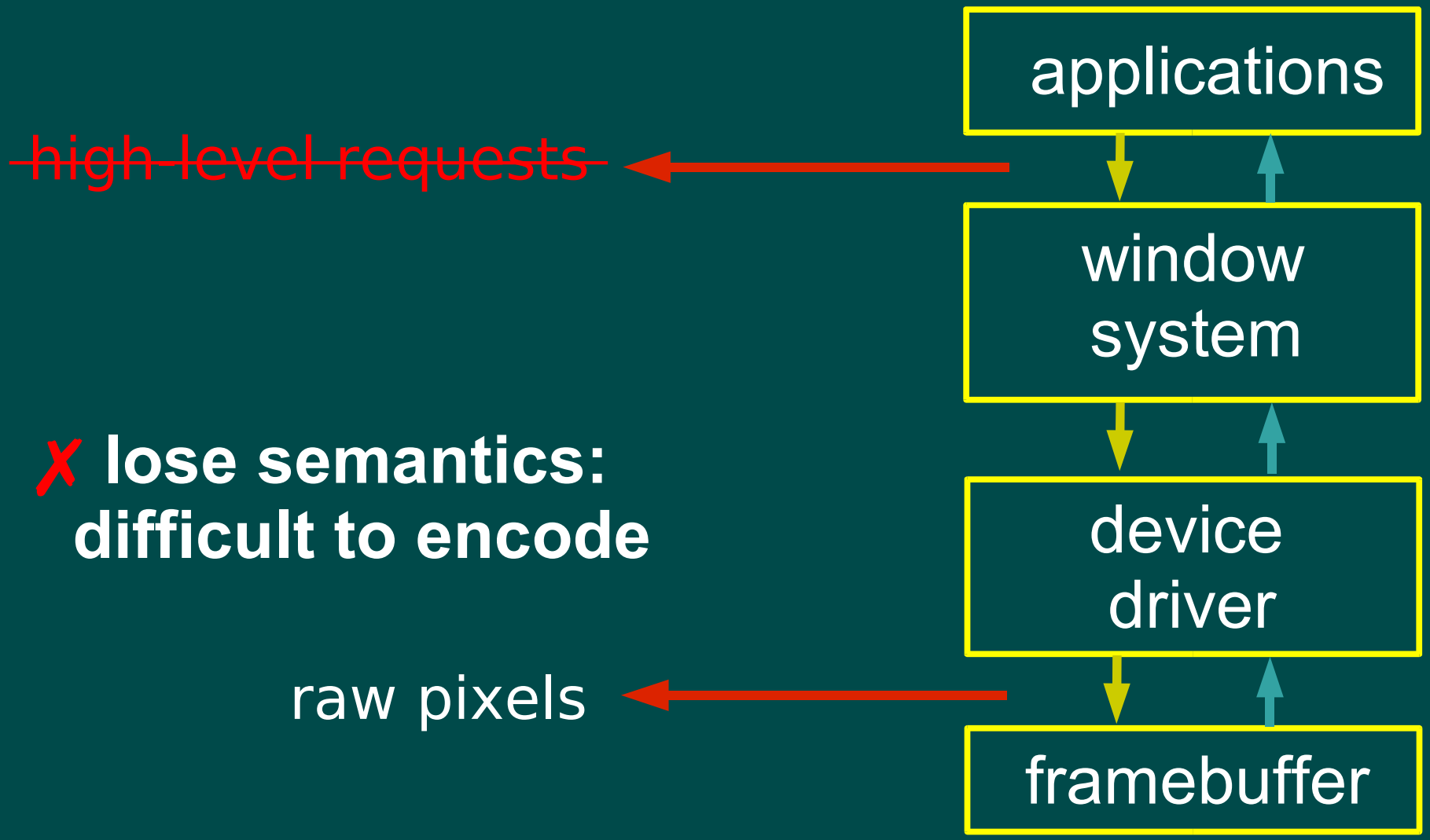
interception and redirection

high-level requests



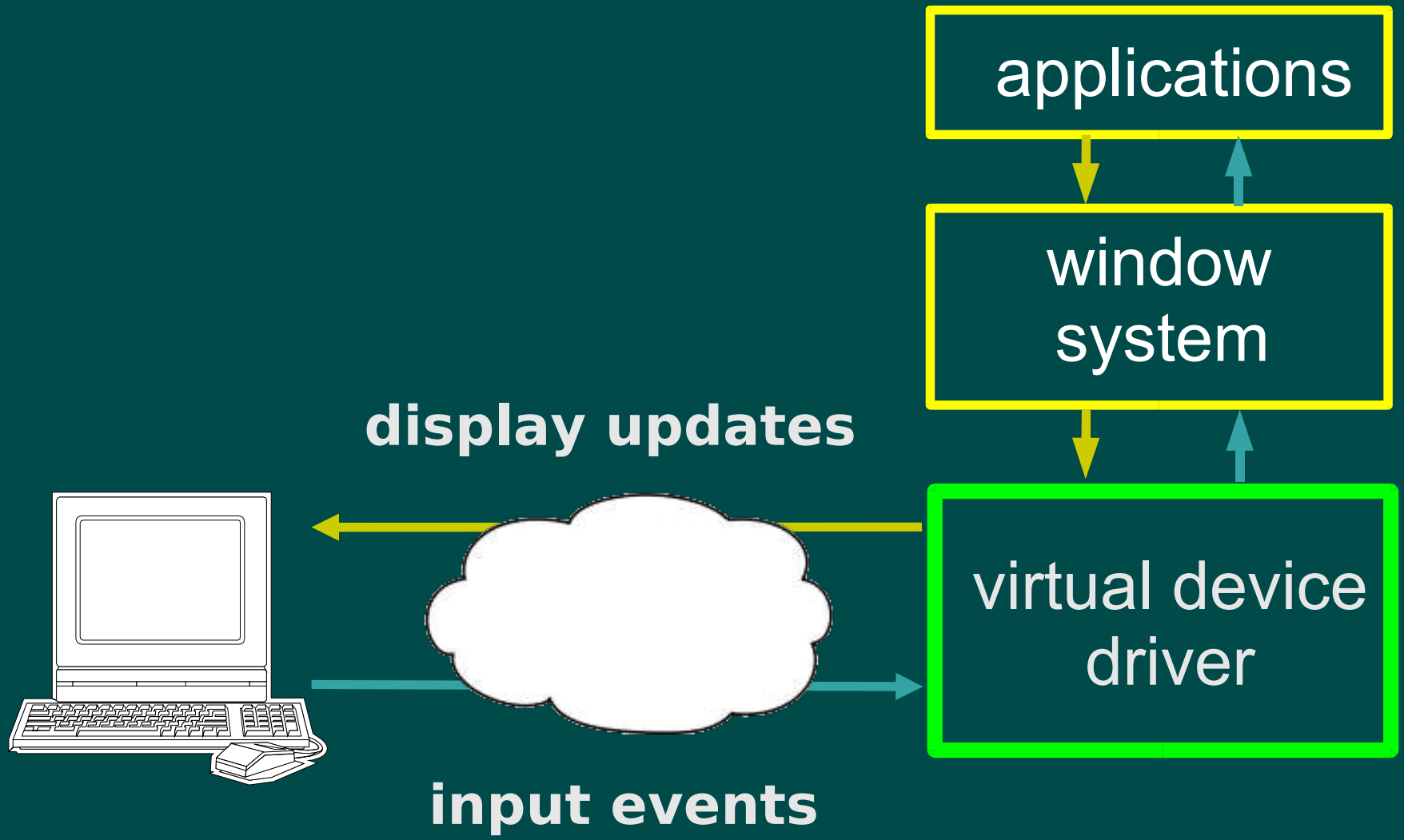
- ✗ **stateful client hurts mobility**
- ✗ **app – window system synchronization**

interception and redirection



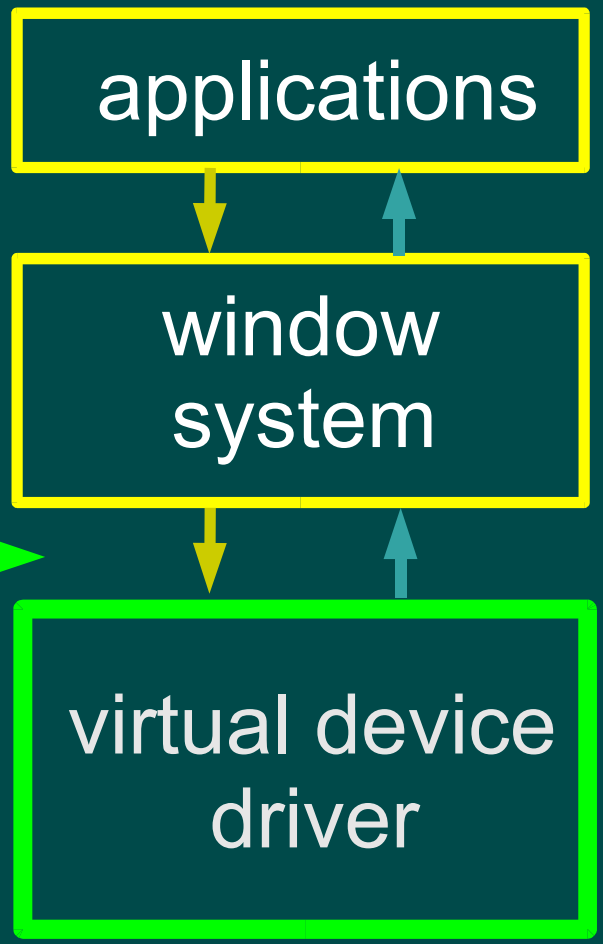
✗ **lose semantics: difficult to encode**

virtual display architecture



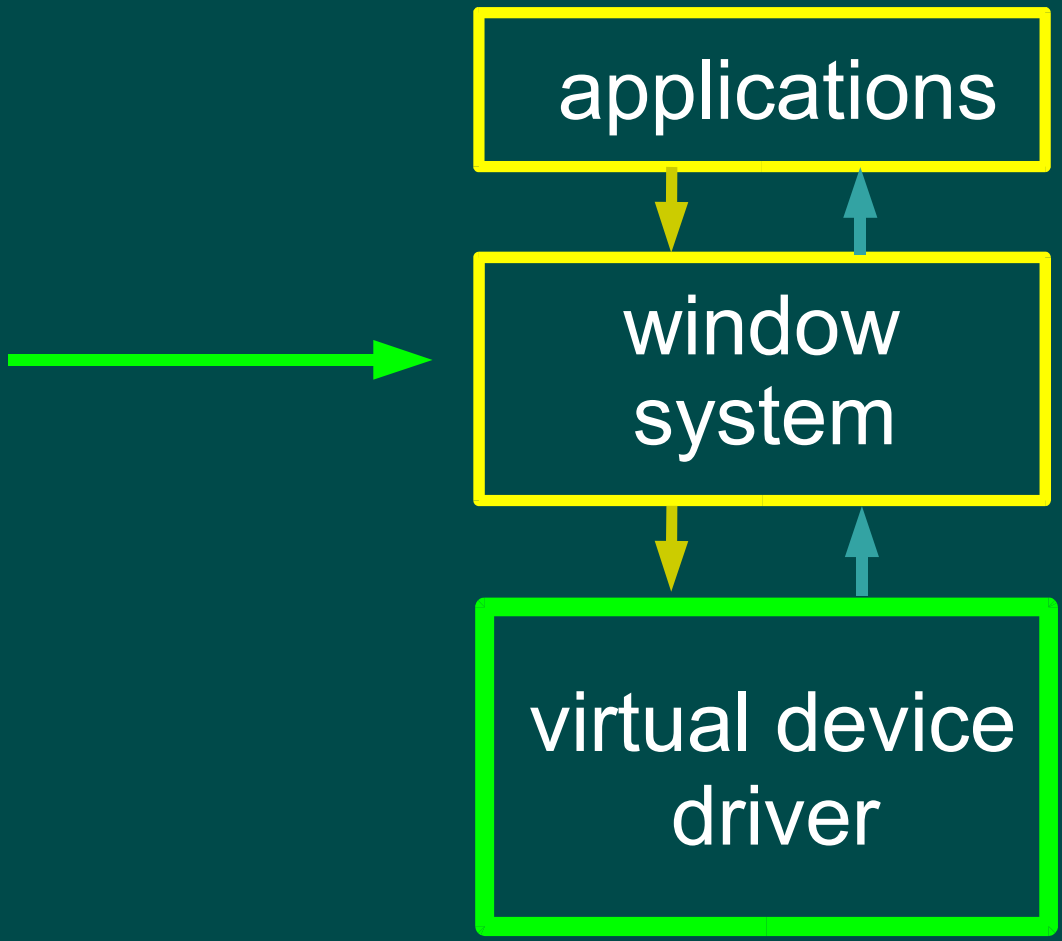
benefits

Standard Interface



benefits

Leverage existing technology



benefits

Simple, low-level protocol

display updates

virtual device driver

benefits

Simple, stateless client

display updates

input events

virtual device driver

display protocol

Inspired by Sun Ray protocol

2D Primitives

- copy
- solid and tile fill
- bitmap fill
- raw

Video

two key problems

how do we translate from application commands to the display protocol?

how and when do we send display updates?

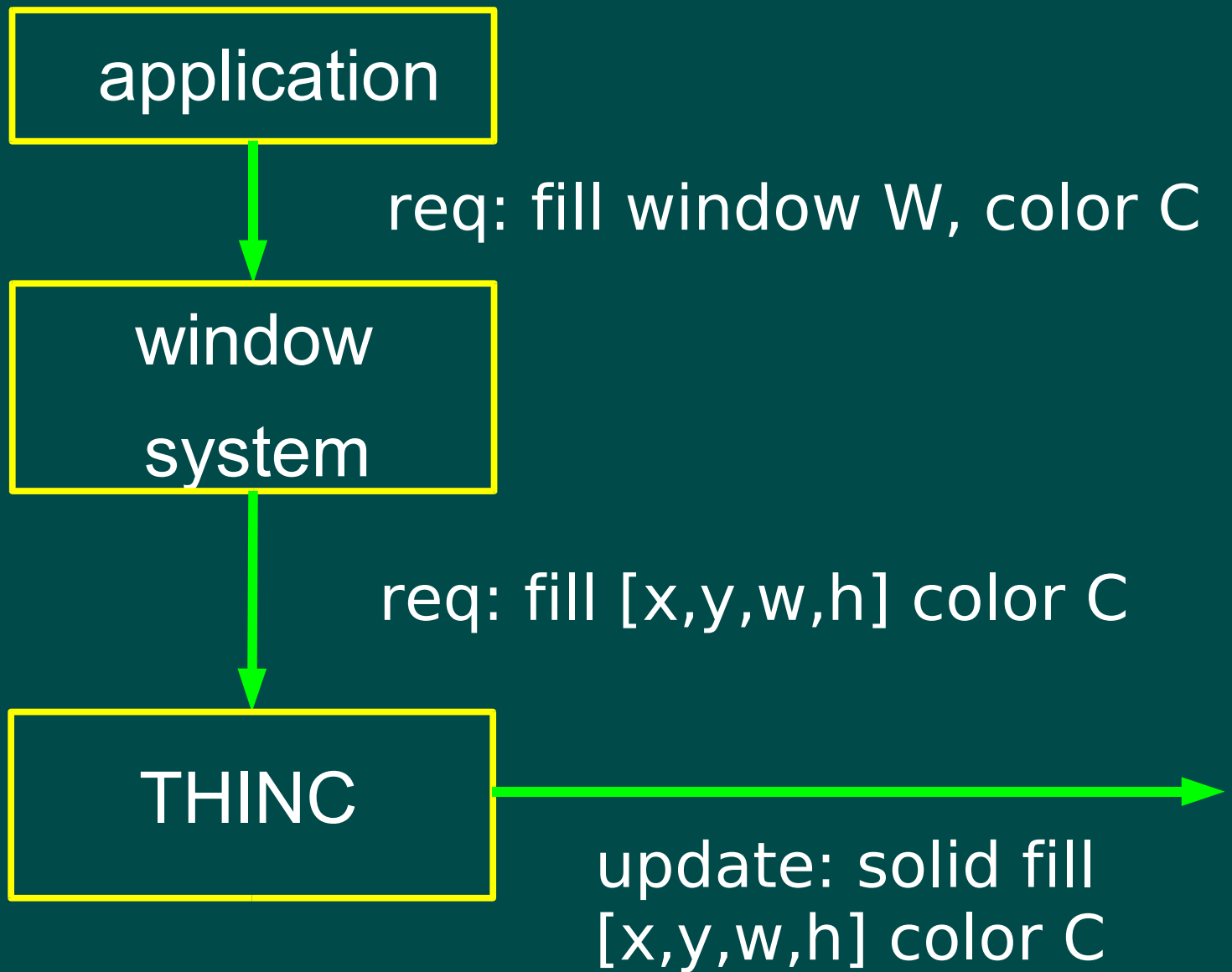
translation

use and preserve semantic information
for efficient translation

translation

- use semantic information when doing translation

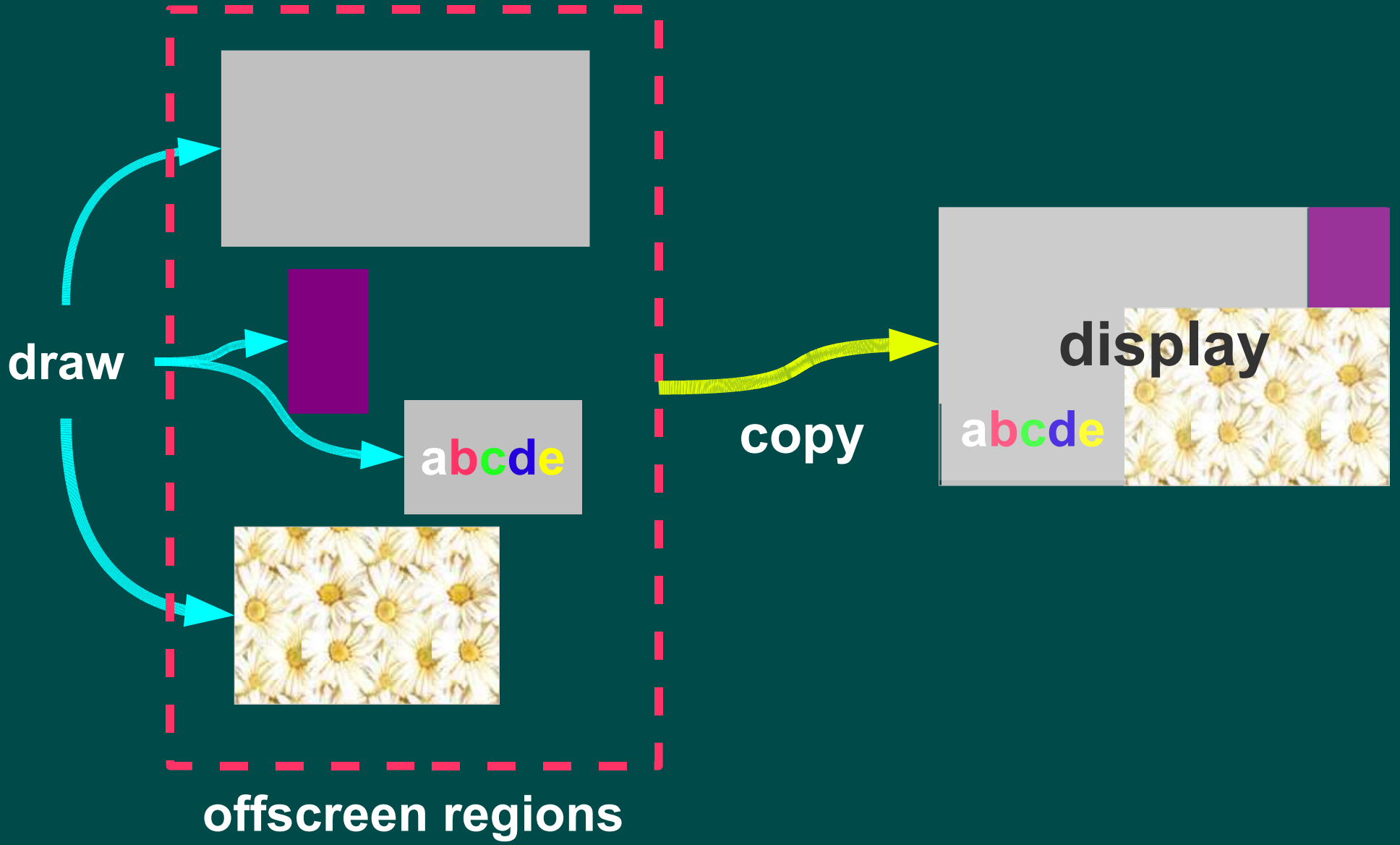
use request semantics to generate update



translation

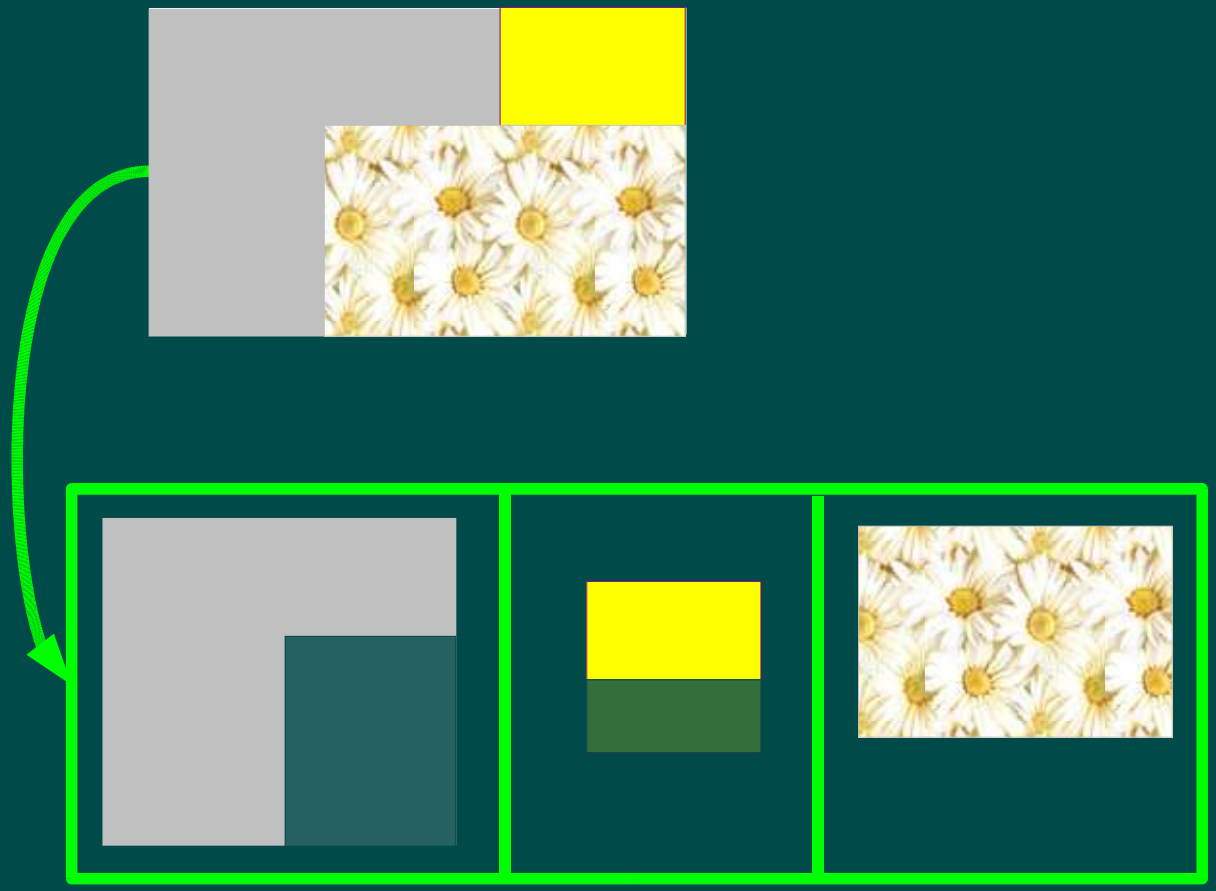
- ✔ use semantic information when doing translation
- preserve semantic information throughout the system

preserving semantics: offscreen rendering



offscreen rendering (cont)

offscreen region



command log

merge, clip, and discard commands as needed

using and preserving semantics: video

- reuse existing hardware acceleration application interfaces
- YUV (luminance-chrominance) color space
  - format independence
  - client hardware acceleration (scaling for free)

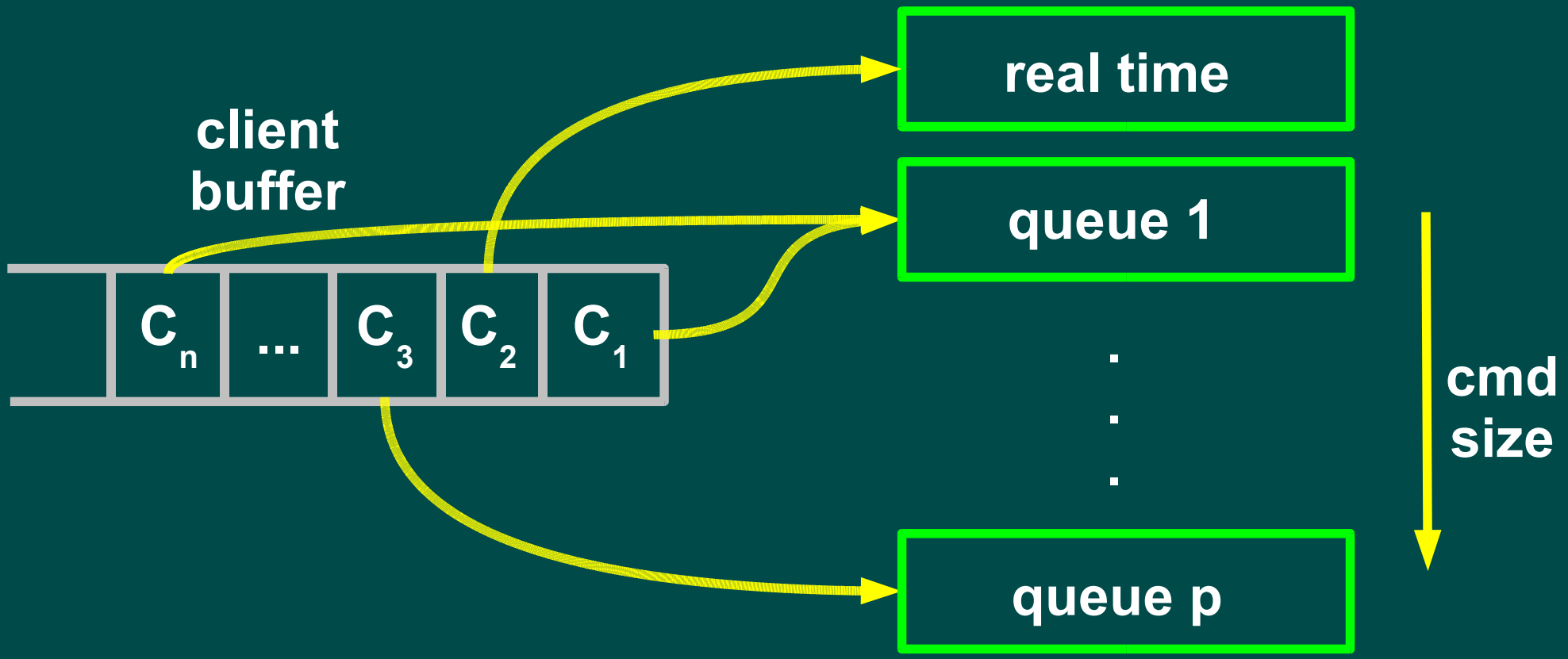
delivery

maximize interactive response of the system

delivery

- transmit updates as soon as possible
- merge, clip, and discard updates as needed

shortest remaining size first scheduler

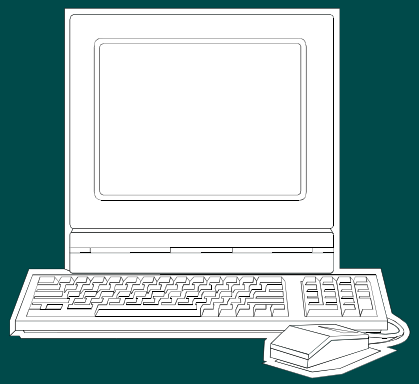


implementation

- X/Linux server
  - ongoing: windows server
- X/Linux, windows, PDA, Java clients

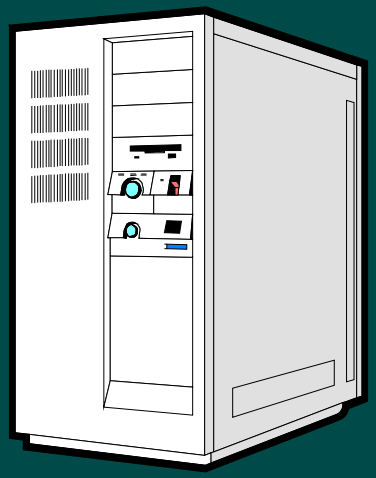
experimental results

- web and video performance
  - comparison to existing systems
  - Internet 2 sites around the globe



LAN

WAN

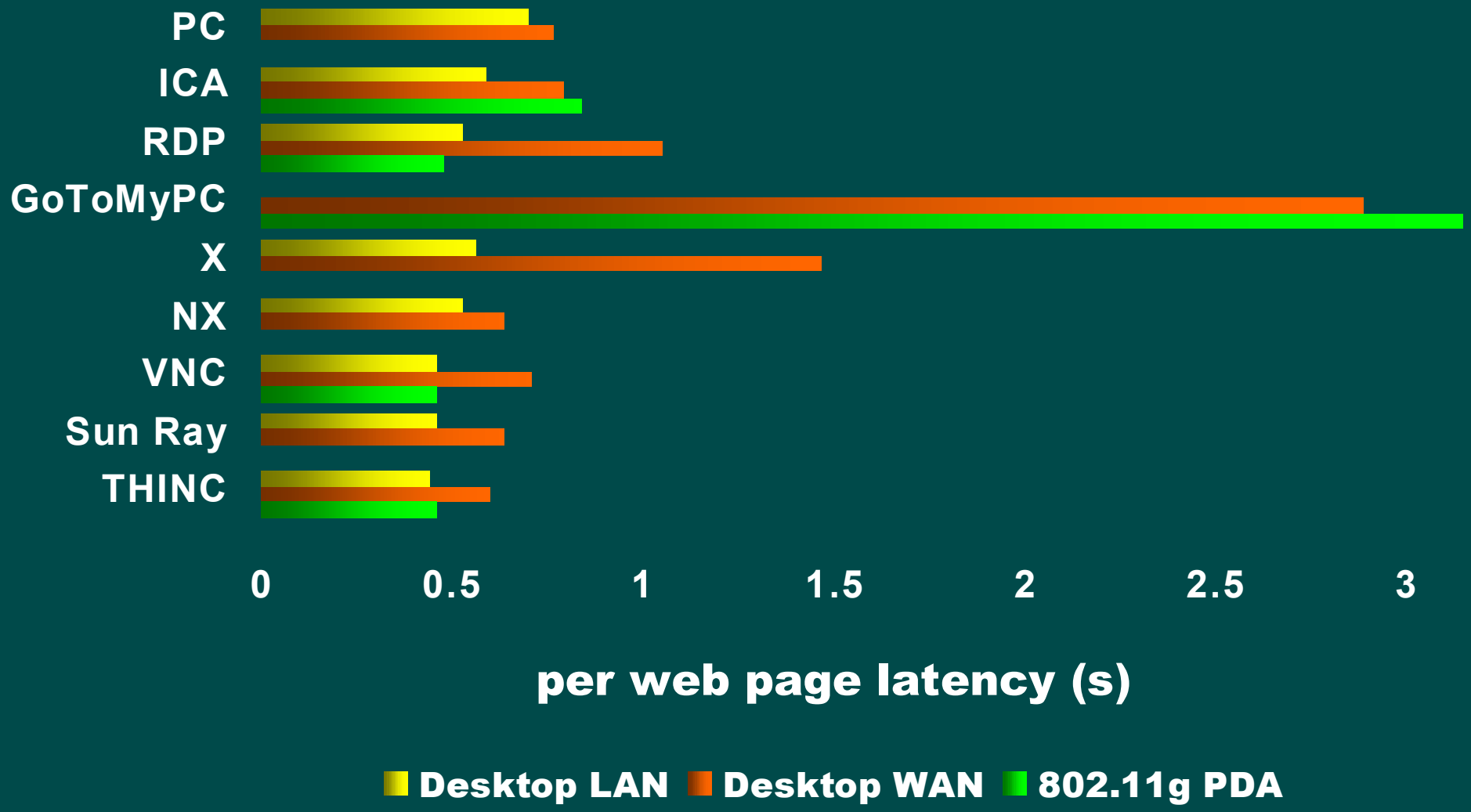


“

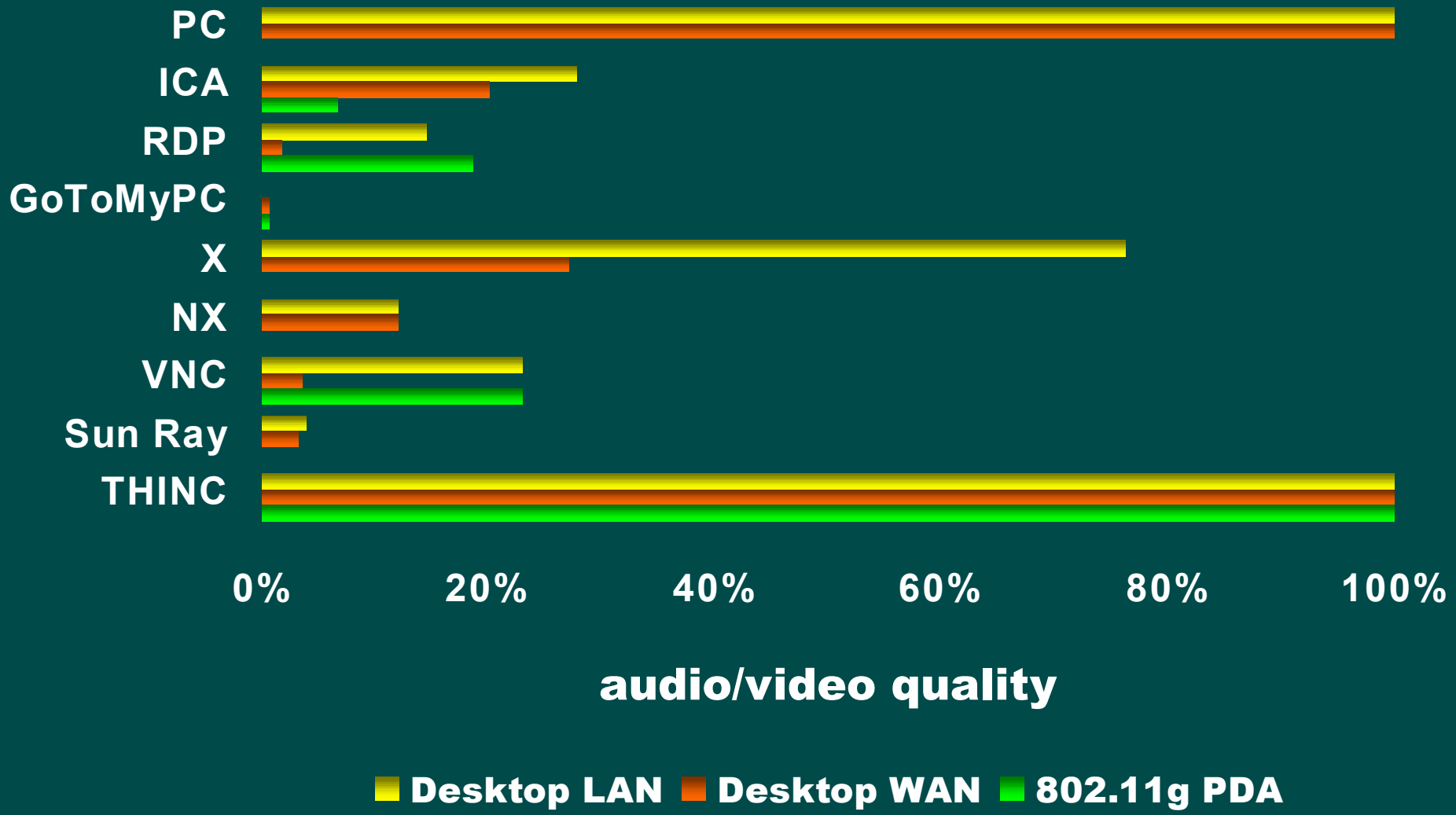
”

802.11g

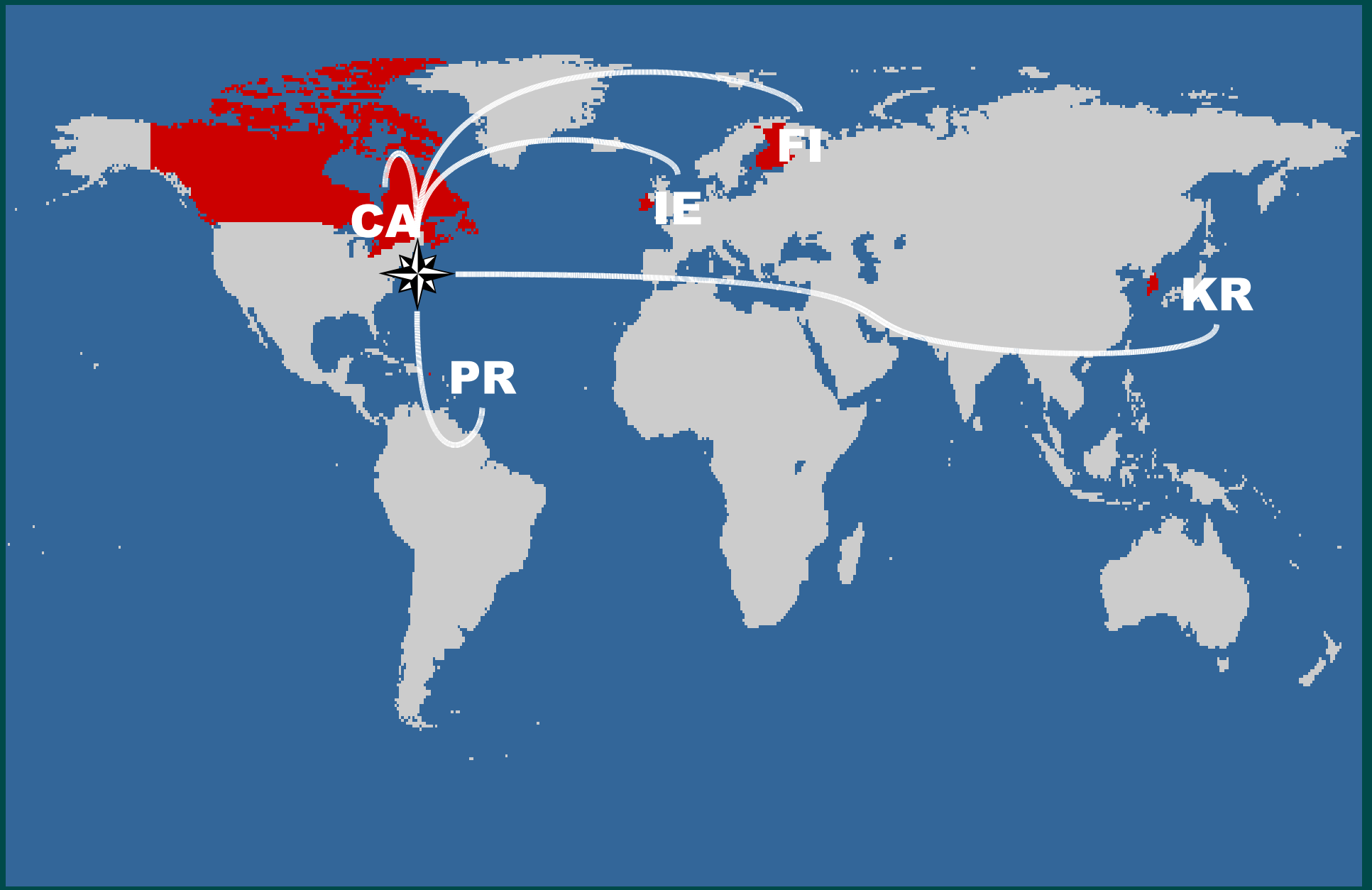
web browsing performance



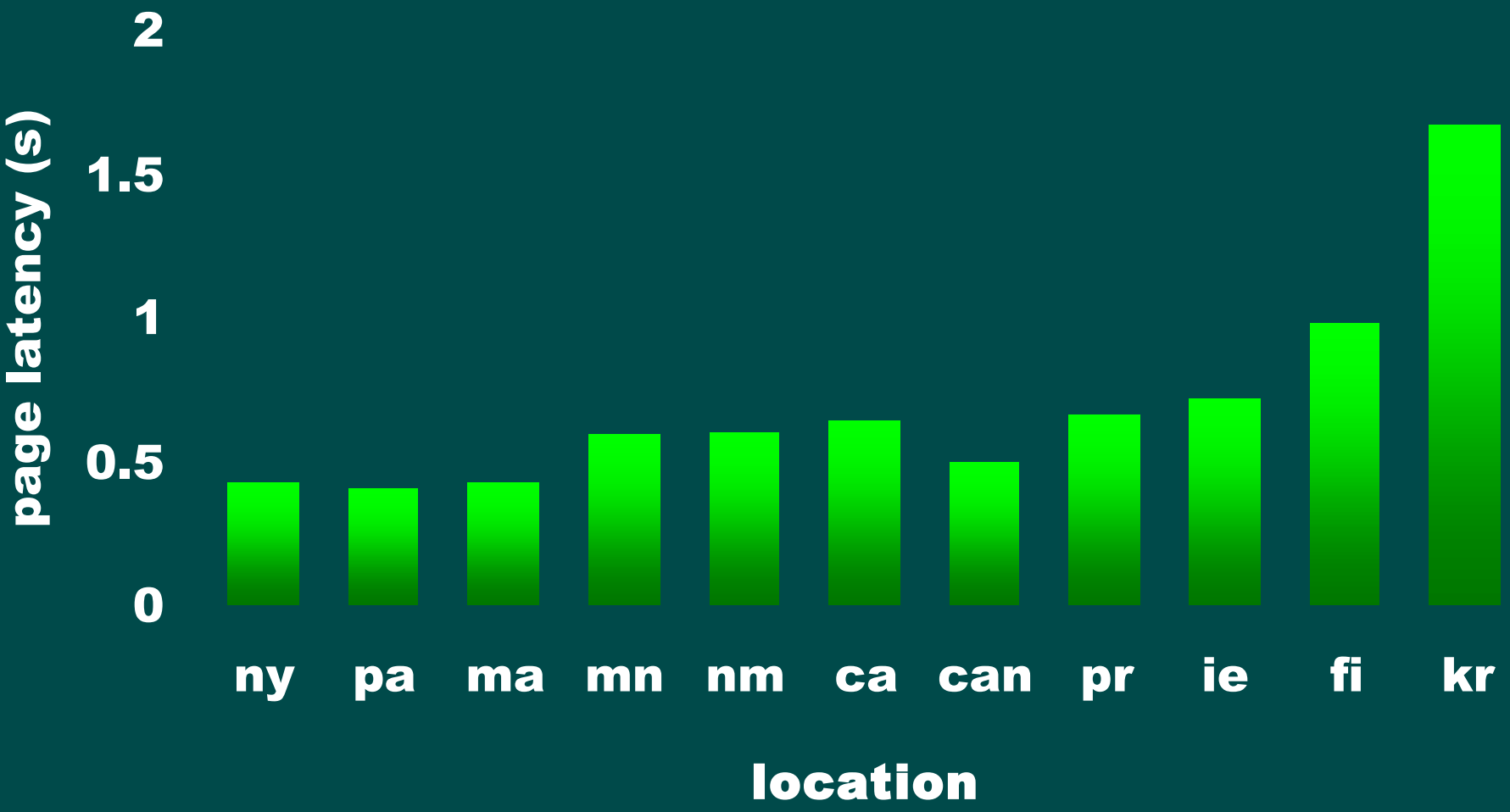
a/v playback quality




MN
MA
PA
NY
CA
NM
not in scale


FI
IE
CA
KR
PR

Internet2 web browsing performance



Internet2 a/v playback quality

demo

conclusions

THINC:

- virtual display architecture transparently leverages existing display infrastructure
- efficient translation by using and preserving semantic information from display request
- delivery mechanisms increase responsiveness of the system

for more info...

http://www.ncl.cs.columbia.edu