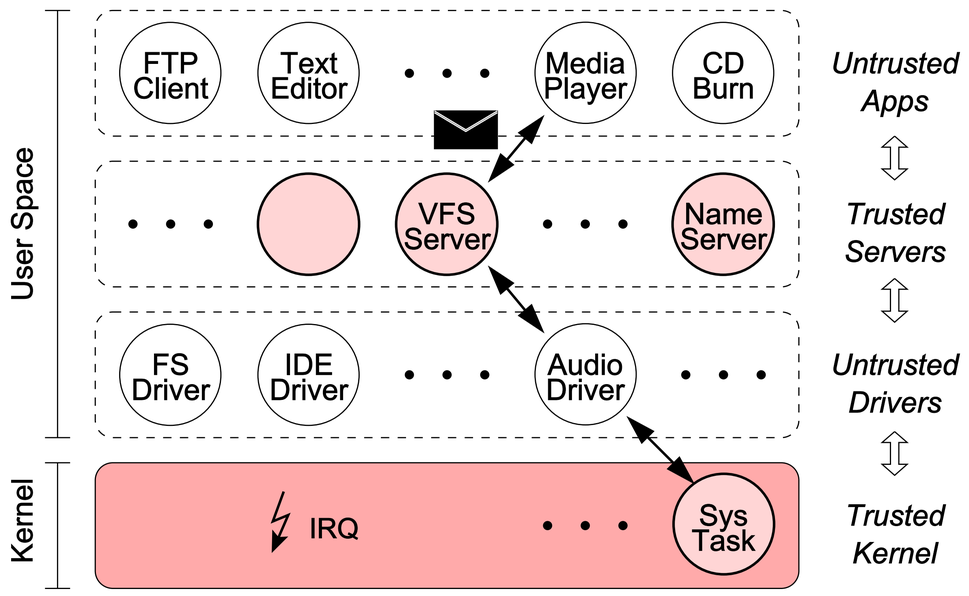
An IPC Model for Extended Asymmetric Trust

Jorrit N. Herder, Herbert Bos, Ben Gras, Philip Homburg, Andrew S. Tanenbaum
{jnherder,herbertb,beng,philip,ast}@cs.vu.nl
Vrije Universiteit, Amsterdam, The Netherlands

IPC in a Generalized Multiserver OS

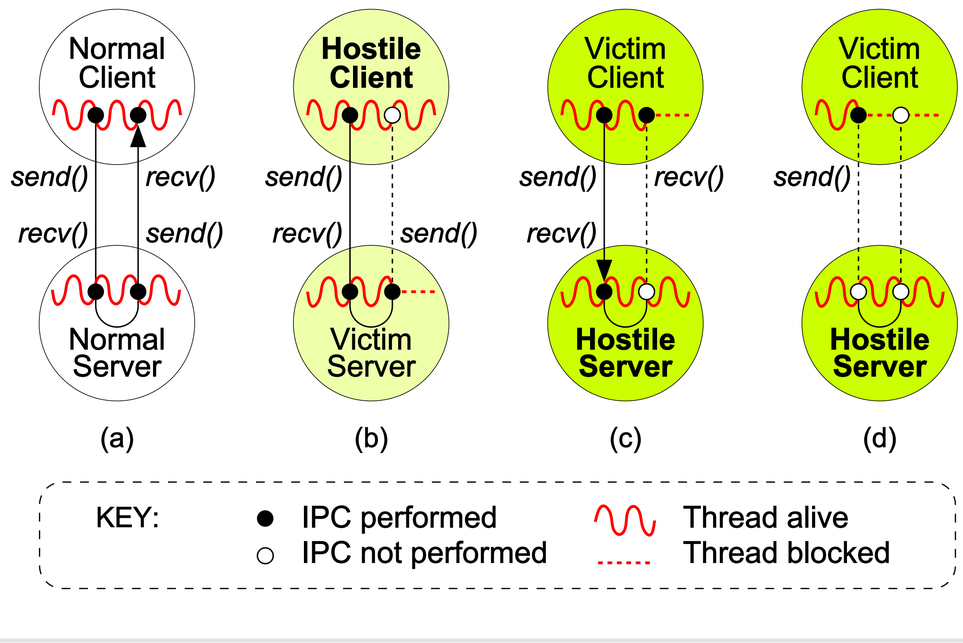
Problem: Dependable IPC infrastructure is a crucial foundation for multiserver operating systems, but the reported vulnerabilities and remedies (Shapiro,2003) do not hold when not only clients, but also servers are considered **unreliable** and **potentially hostile**.



This generalized OS model is followed by MINIX 3, but also applies to commodity systems like Windows

New Vulnerabilities in Synchronous IPC Designs

Insight: Drivers acting as servers to components at a higher-level are a **realistic threat** to dependability, as they typically comprise 70% of the OS and have error rates 3-7x higher than other code. Experiments show rogue drivers indeed can easily hang the OS.



Recent SWIFI experiments with our Ethernet drivers caused system-wide hangs within seconds

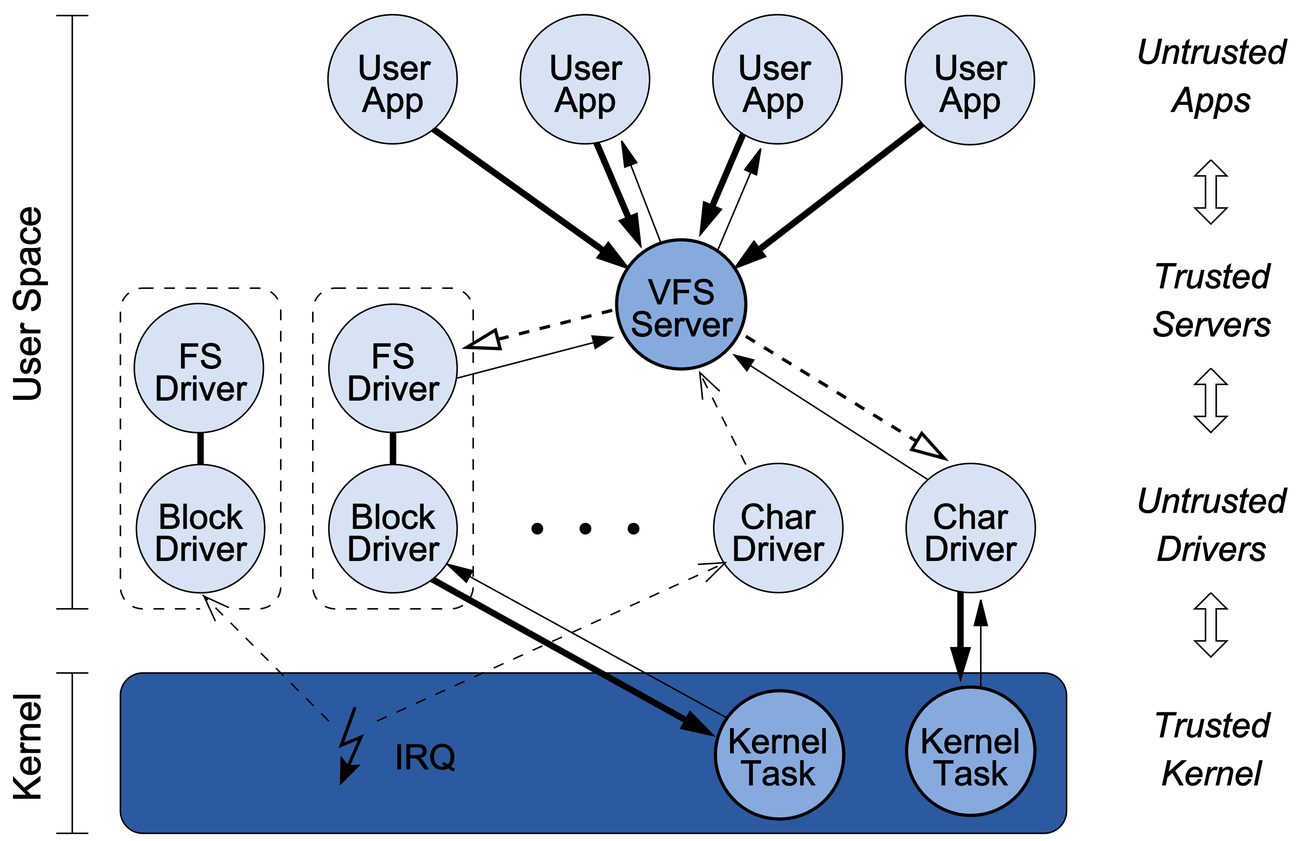
Dependable IPC Design as Implemented in MINIX 3

SENDREC
Restrict **untrusted clients** to fully synchronous rendezvous

ASEND
Use asynchronous send to contact **untrusted servers**

NBSEND
Allow unbuffered, nonblocking IPC for reply messages

NOTIFY
Support single-bit notifications to signal system events



The guiding principle in MINIX 3 is to keep it simple. Therefore, we based our IPC design on three architectural constraints: no multithreading, no timeouts, and no demand paging. We also tried to keep the programming model straightforward.

Supported by the Netherlands Organization for Scientific Research (NWO) under grant 612-060-420.