Systems Software Research Group
http://ssrg.ece.vt.edu

Multi-Variant Execution atop a Decomposed Hypervisor on Emerging Heterogeneous-ISA Multicore

Pierre Olivier, Antonio Barbalace, Binoy Ravindran
Systems Software Research Group, ECE Department, Virginia Tech, USA
{polivier, antoniob, binoy}@vt.edu

Supported by:

Goal: Improve Software Security in the Cloud

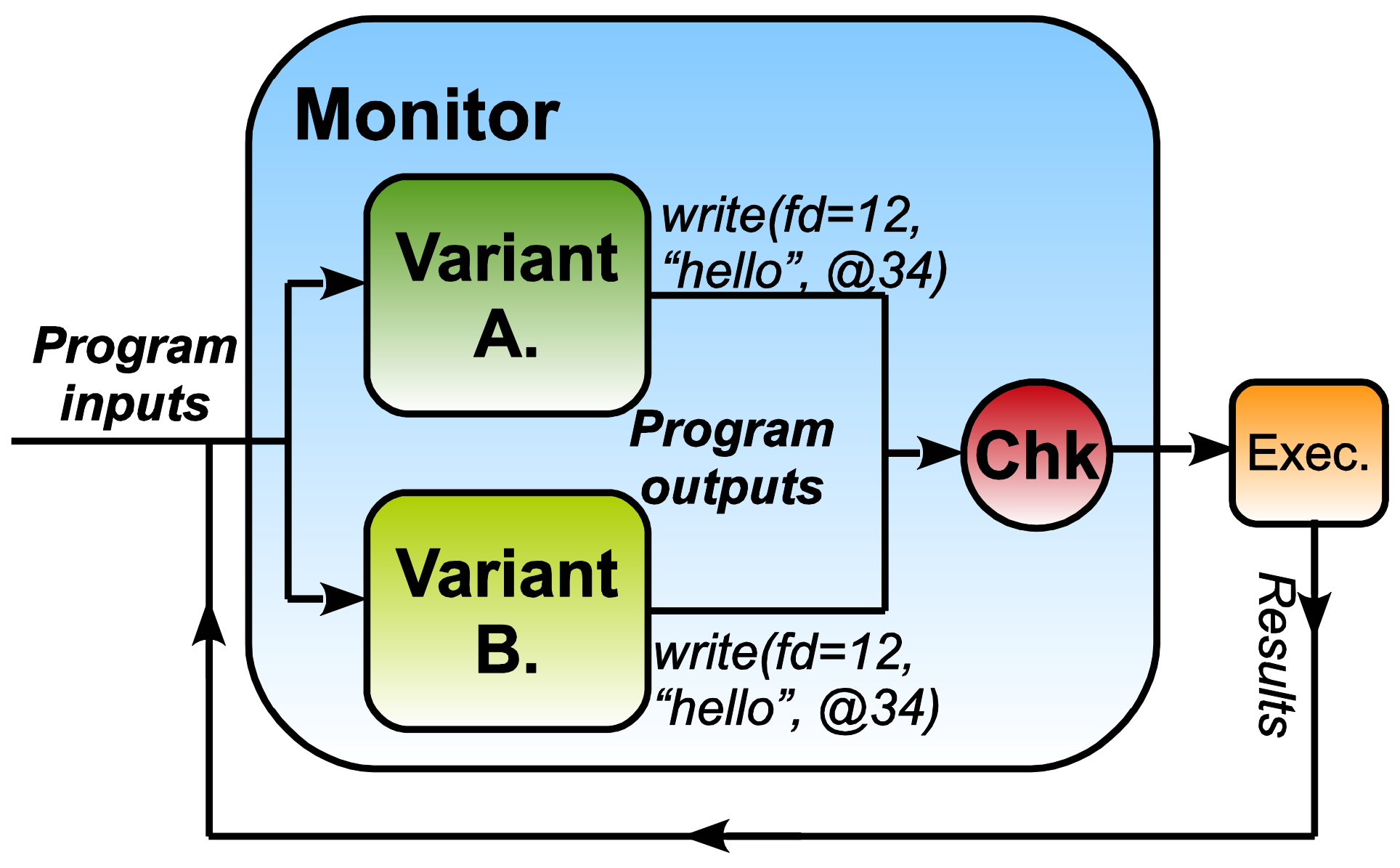
- **Major concern** in a multi-tenant environment
- **Multi-variant execution** [1] (**MVX**) protects regular processes against **control flow diversion** attacks leading to arbitrary code execution (e.g., buffer overflows)
- **Proposition: adapt MVX to virtualization for the *Xen* hypervisor on heterogeneous multicores**
- MVX for **virtual machines** and **hypervisor components**
- **New type of variance: ISA difference**

Xen

- Popular in cloud environments (Amazon, Rackspace, etc.)
- Bare-metal: **isolation** [2]
- **Unikernels** [3]: security oriented, single purpose applications running as guest VMs

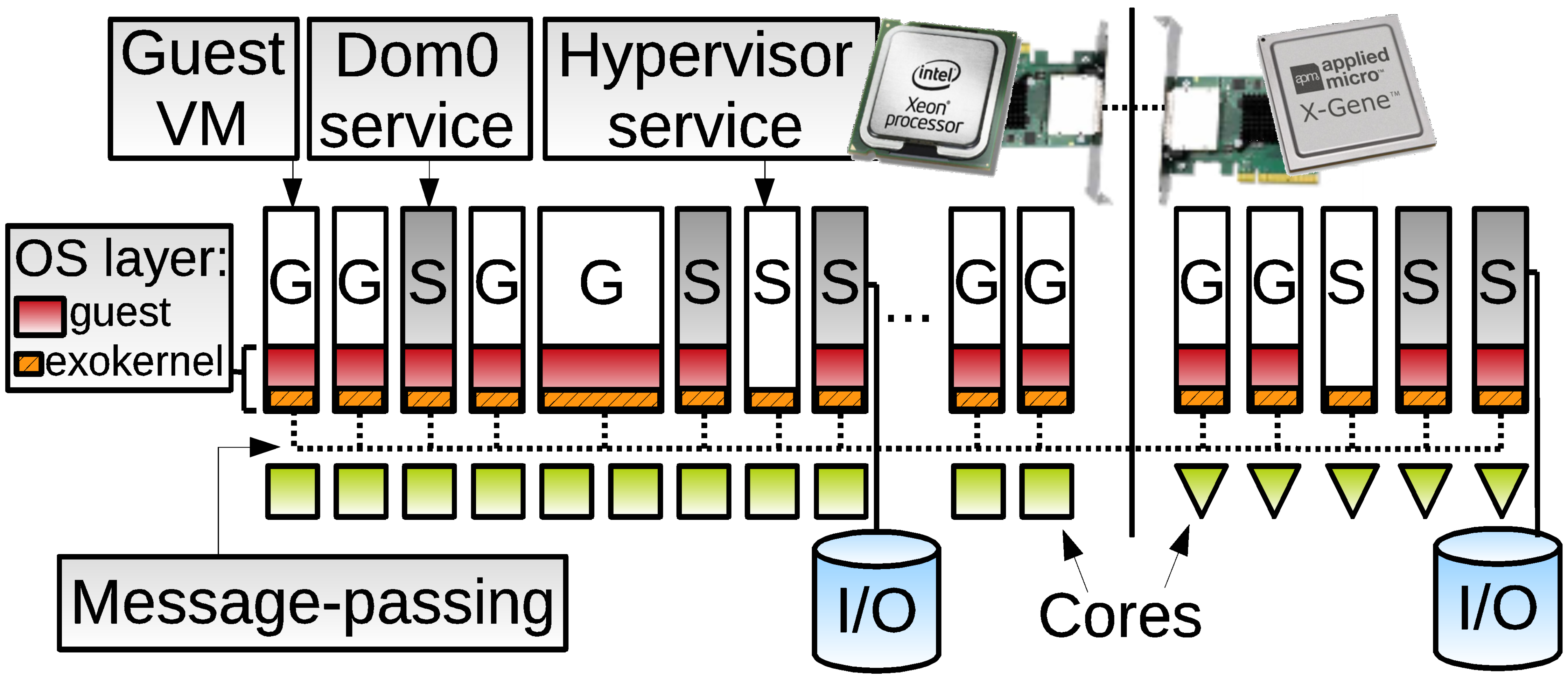
Multi-Variant Execution [1]

- Instances (**variants**) of the same program run in parallel and in lockstep mode
- Abstracted by a **monitor** distributing inputs, and comparing outputs
- Variants are semantically equivalent and structurally different: They **react differently in the case of an attack** (e.g., reverse stack grow)

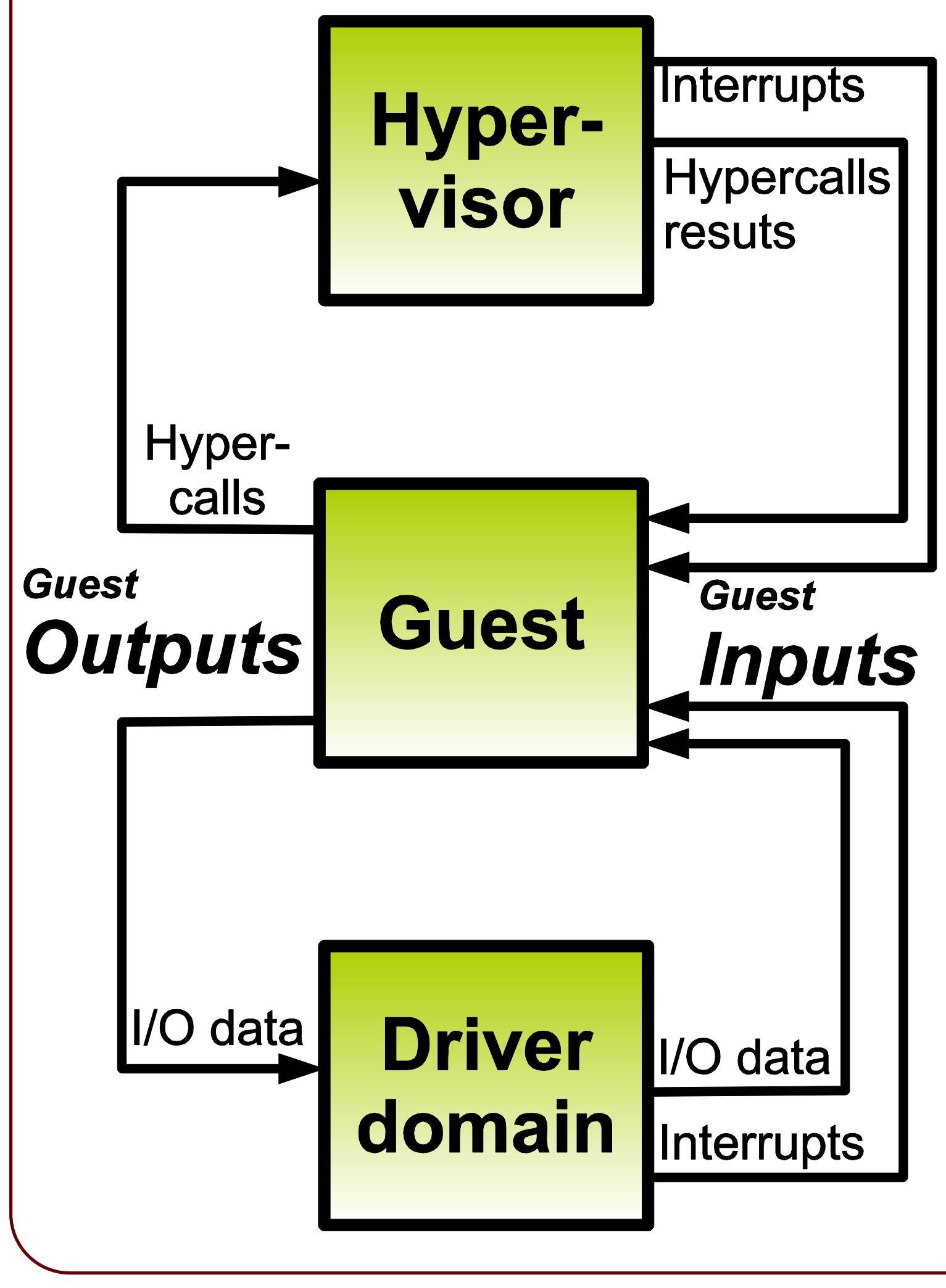


Broken-Hype: Decomposed Virtualization Services

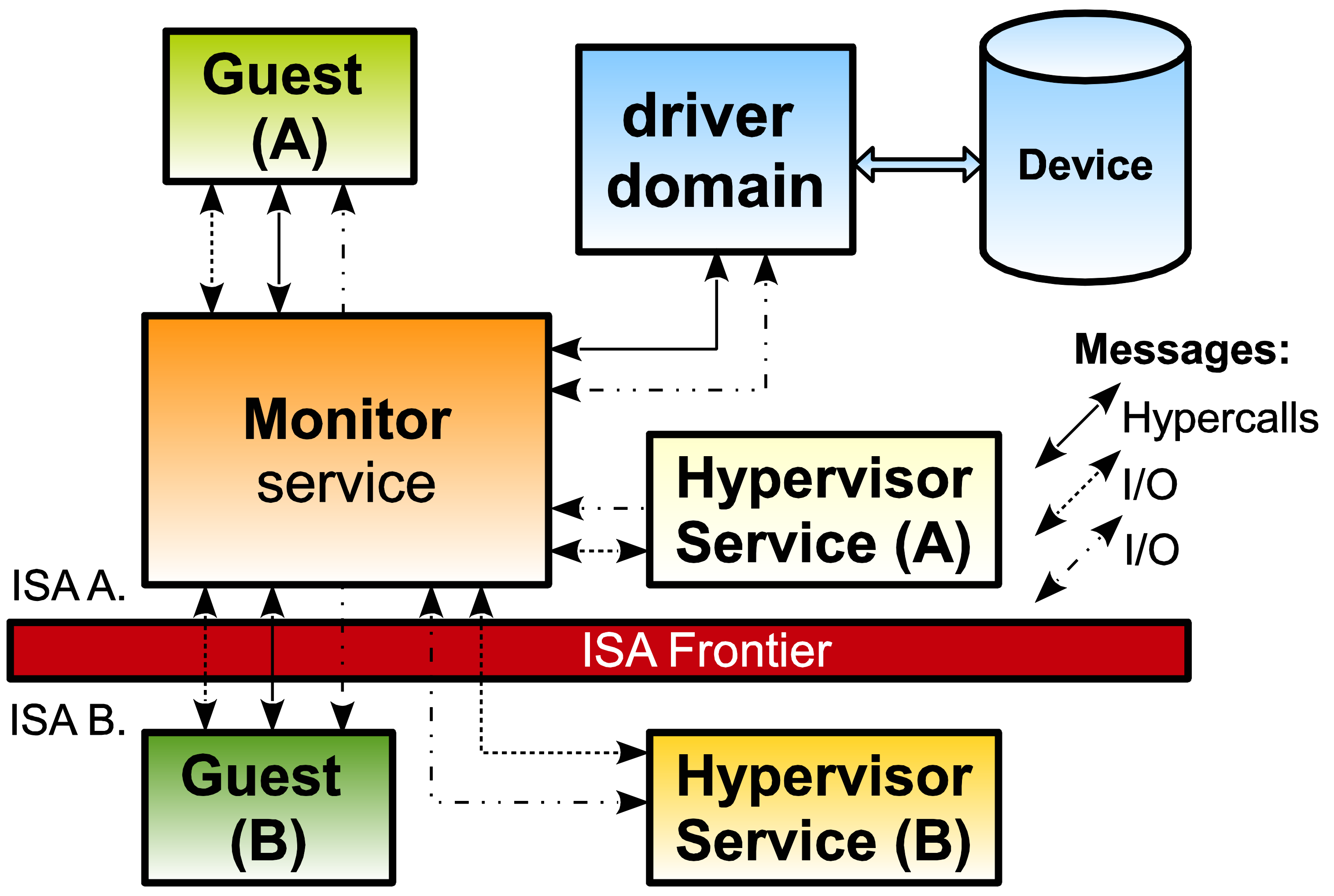
- Xen: **Hypervisor** + **Privileged VM** (*Dom0*)
  - Complex for MVX in their current state
- **Decompose virtualization layer into independent services → MVX for this layer:**
  - VM management, VM boot control, Xenstore, drivers, etc.
- **Small exokernel with min. functionalities:**
  - **Messaging layer** for communication, **Interaction between components: message**
    - Supports **crossing ISA boundaries**
  - TCB: exokernel + small hypervisor



Multi-Variant Execution of Virtualization Components



- **Granularity of checks: message → Hypercalls, interrupts, and I/O**
- **Monitor intercepts all messages sent / received by entities in MVX**
- ISAs difference: **more variance & diversity**
- **Inconsistencies / false positives:**
  - Non-immutable results, Interrupts distribution → **monitor intervention**
  - Scheduling: VCPUs and Multi-threading inside a guest → **deterministic multi-threading, *replicatable* determinism** [4]



Conclusion

- Proposal: ***adapt MVX to the virtualization world*: MVX for guest VMs and for the virtualization layer itself** (Hypervisor and control VM)
- ***Broken-Hype*, decomposed virtualization layer design: isolated components, minimal exokernel, message passing communication**
- New type of variance: ***ISA difference* → strong variance and diversity**

References

[1] B. Cox and E. David "N-variant systems: a secretless framework for security through diversity." Usenix Security. Vol. 6. 2006.
[2] P. Colp et al. "Breaking up is hard to do: security and functionality in a commodity hypervisor." ACM SOSP, 2011.
[3] A. Madhavapeddy et al. "Unikernels: Library operating systems for the cloud." ACM SIGPLAN Notices. Vol. 48. No. 4. ACM, 2013.
[4] S. Volckaert et al. "Replicatable determinism for parallel programs." 2015.