Deep Specifications and Certified Abstraction Layers

Ronghui Gu, Jérémie Koenig, Tahina Ramananandro, Zhong Shao, Xiongnan (Newman) Wu, Shu-Chun Weng, Haozhong Zhang, Yu Guo

Yale University

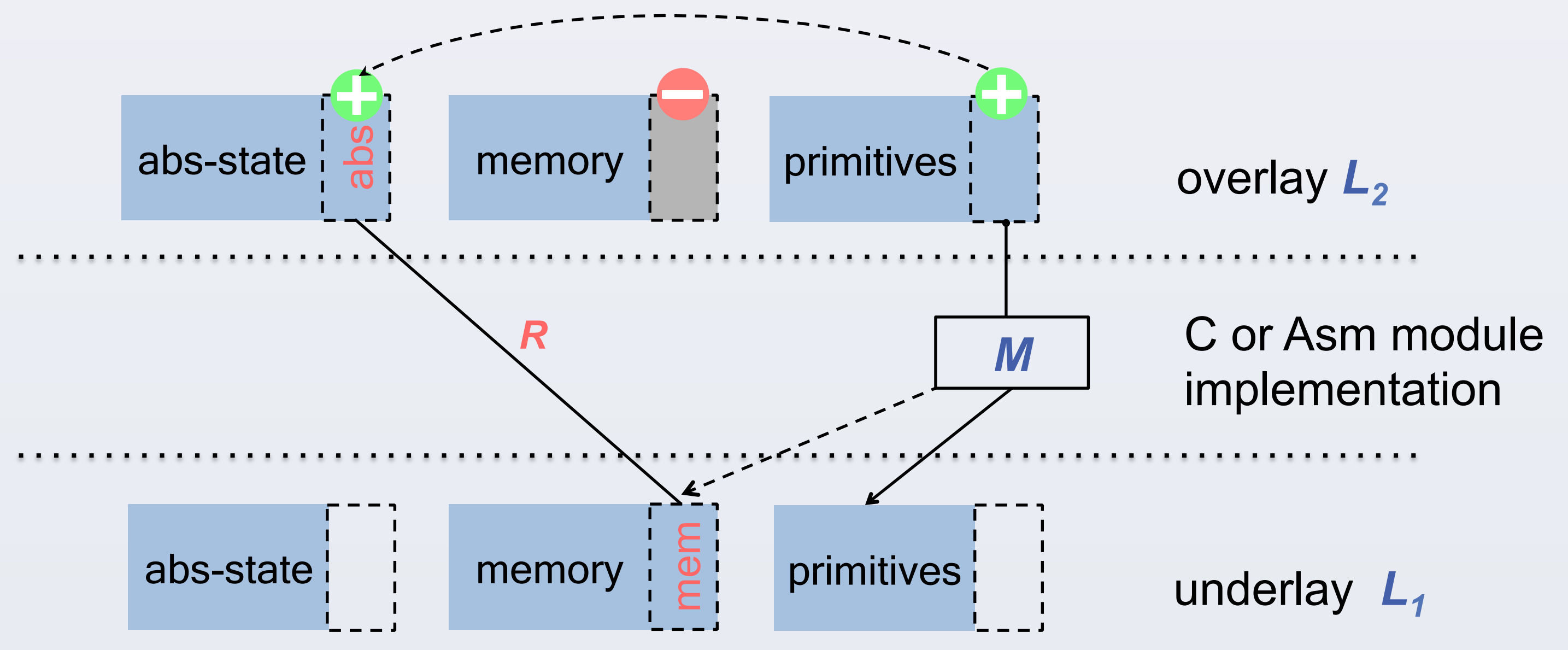
Artifact • Consistent • Complete • Well Documented • Easy to Reuse • Evaluated • AEC • POPL

Contributions

1. Present the first language-based account of certified abstraction layers and show how they correspond to a rigorous form of abstraction over deep specifications used widely in the system community.
2. Provide a layer calculus showing how to formally specify, program, verify, and compose certified abstraction layers.
3. Instantiate the layer calculus on top of two core languages: *ClightX*, a variant of the CompCert Clight language; and *LAsm*, an x86 assembly language.
4. Extend CompCert to build a new verified compiler, CompCertX, that can compile *ClightX* abstraction layers into *LAsm* layers.
5. Construct several feature-rich certified OS kernels in Coq. The hypervisor consists of 5500 lines of C and x86 assembly, and can boot a version of Linux as a guest.

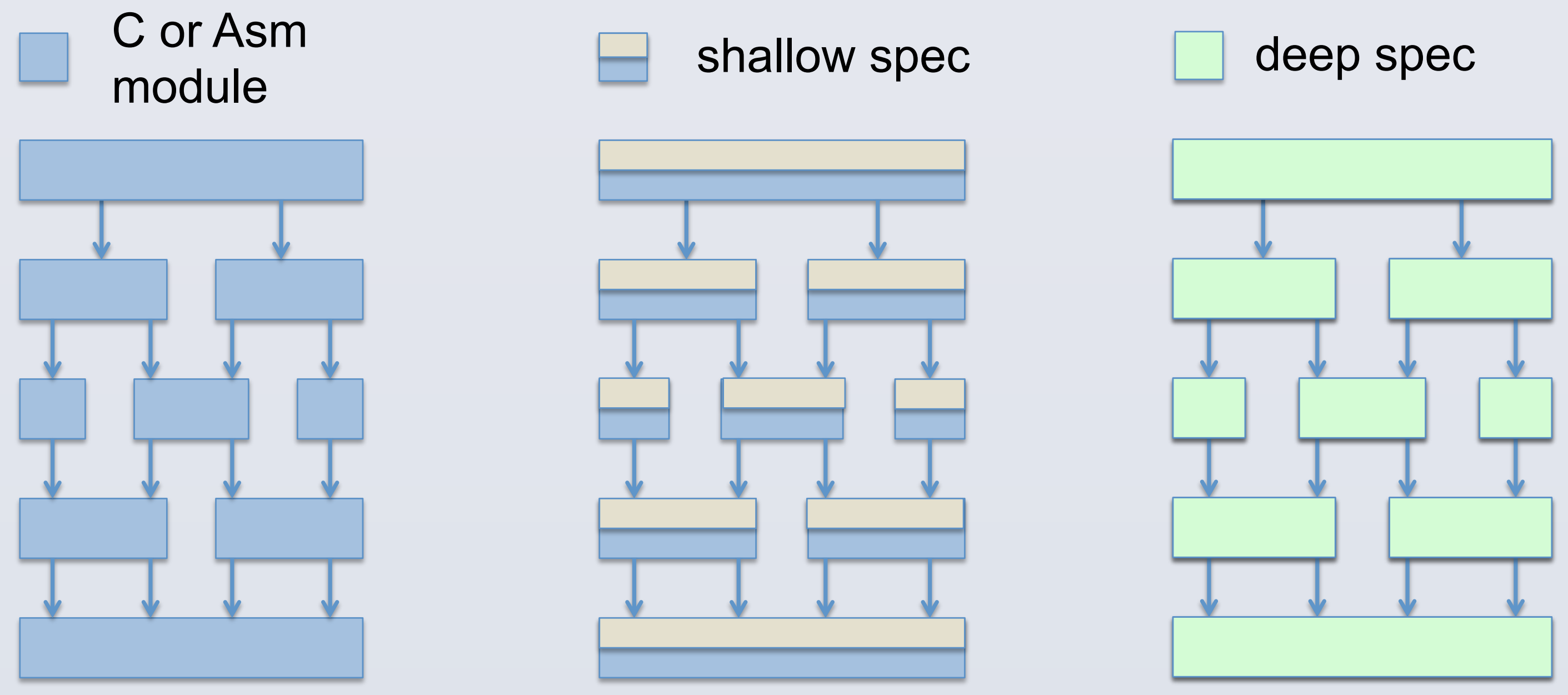
Abstraction Layer

- An abstraction layer is a triple (L_1, M, L_2).
- The module M implements the overlay interface L_2 on top of underlay L_1.

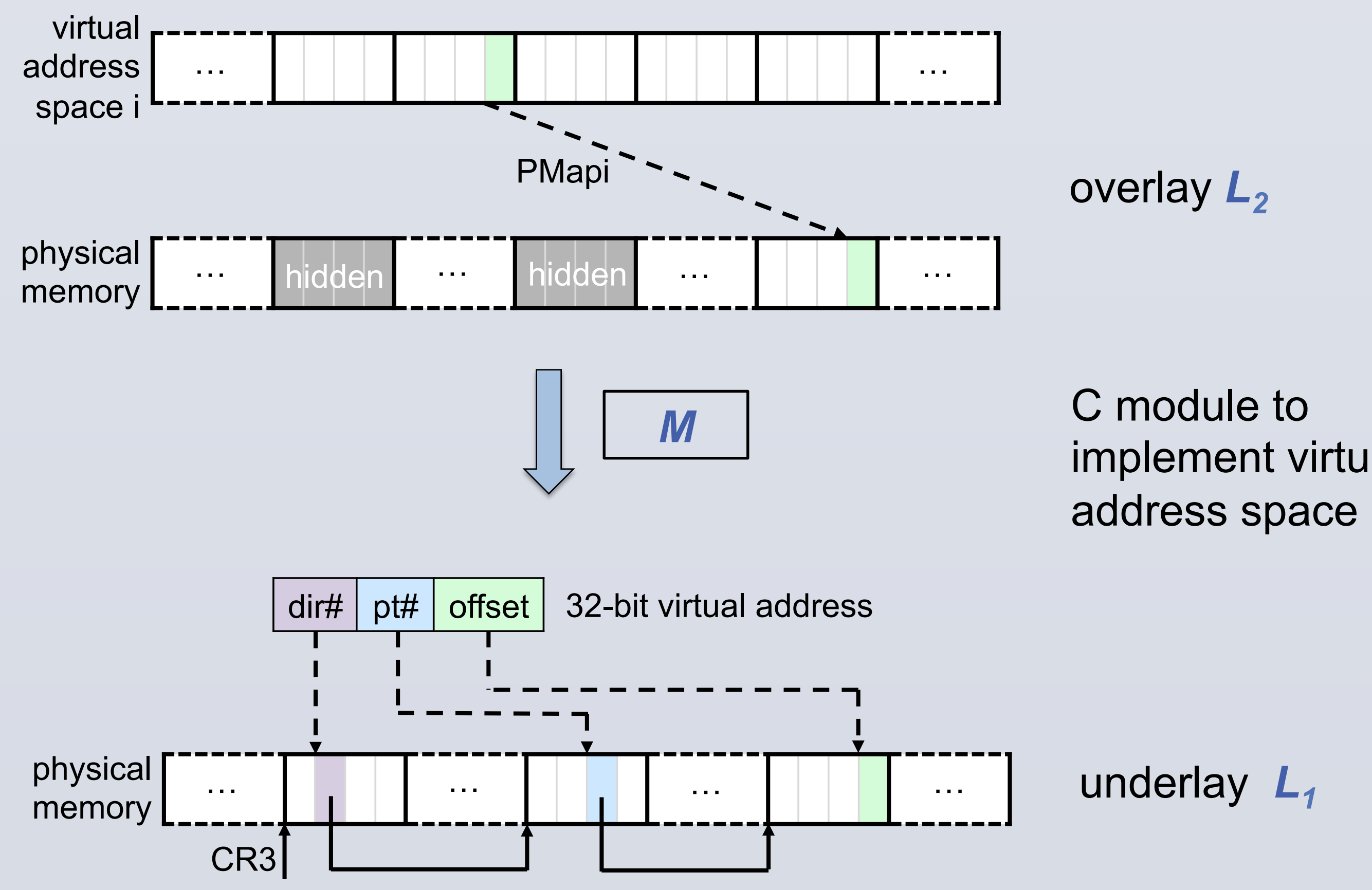


Deep Specification

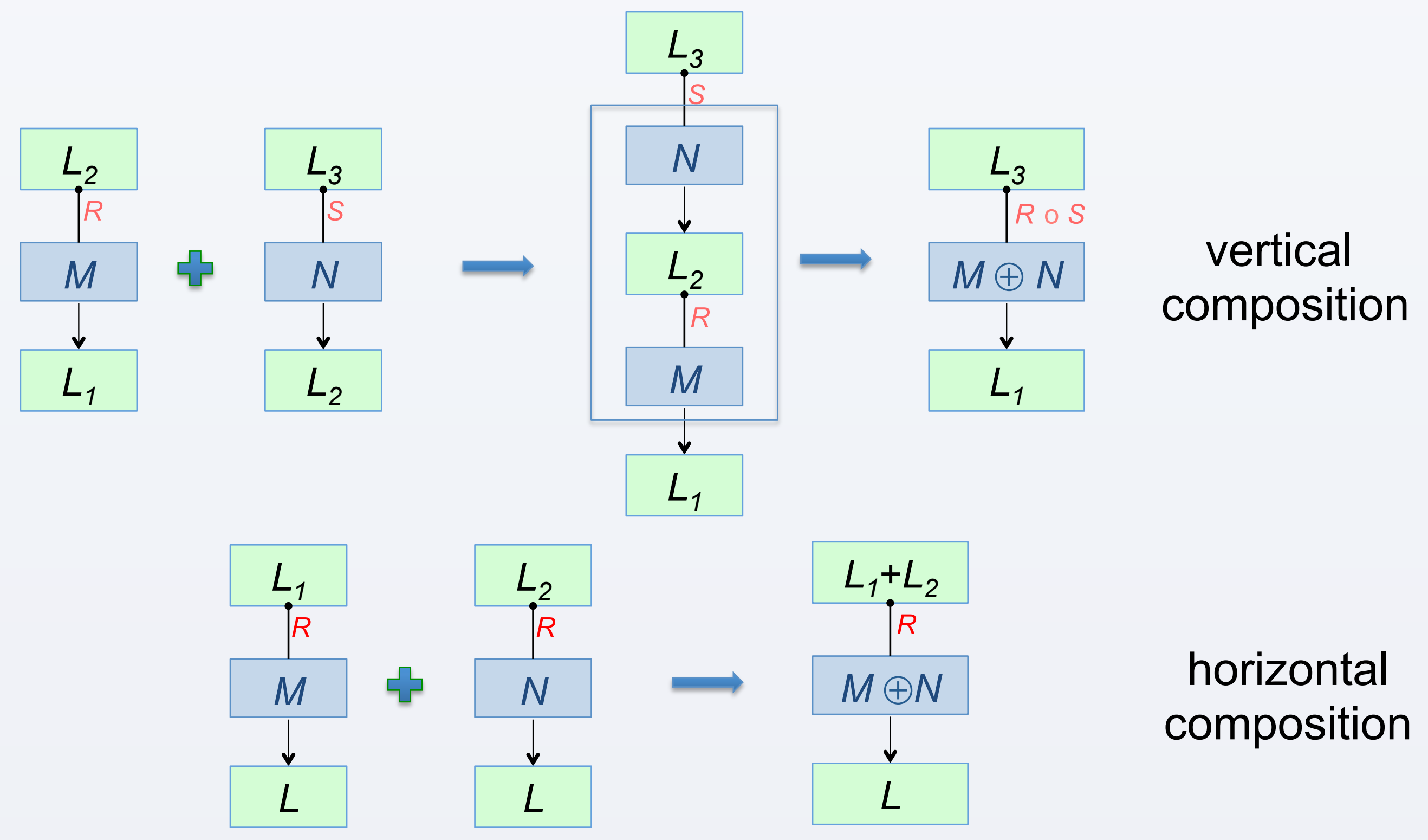
- L_2 is a deep specification of M over L_1 if under any valid program context P of L_2, the whole-program semantics 【 $P \oplus M$ 】 (L_1) and 【P】 (L_2) are observationally equivalent.
- Deep specification captures all we need to know about a module *M*.
- Any two implementations of the same deep spec are contextually equivalent.



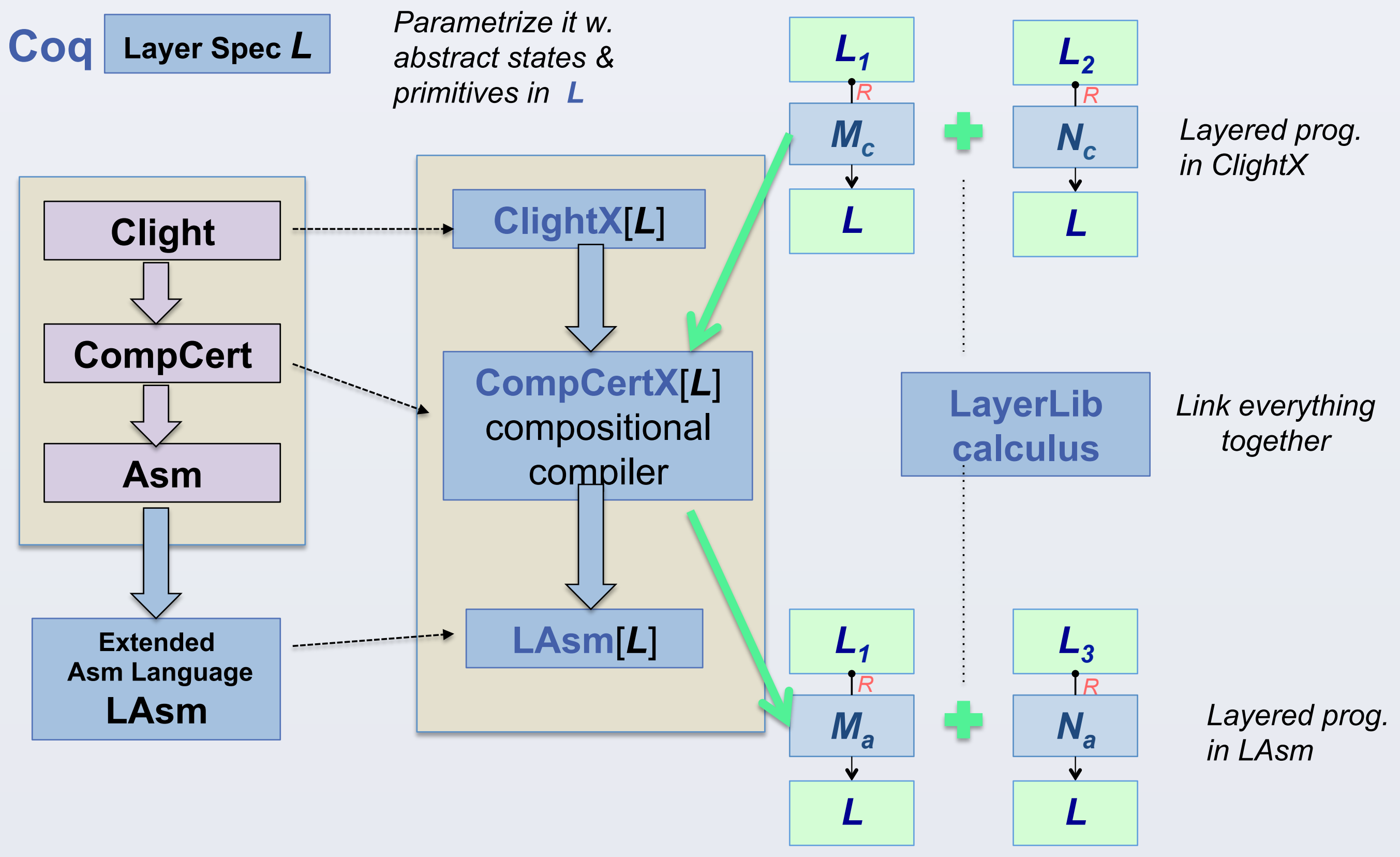
Example: Page Map and MemoryMode



Layer Calculus



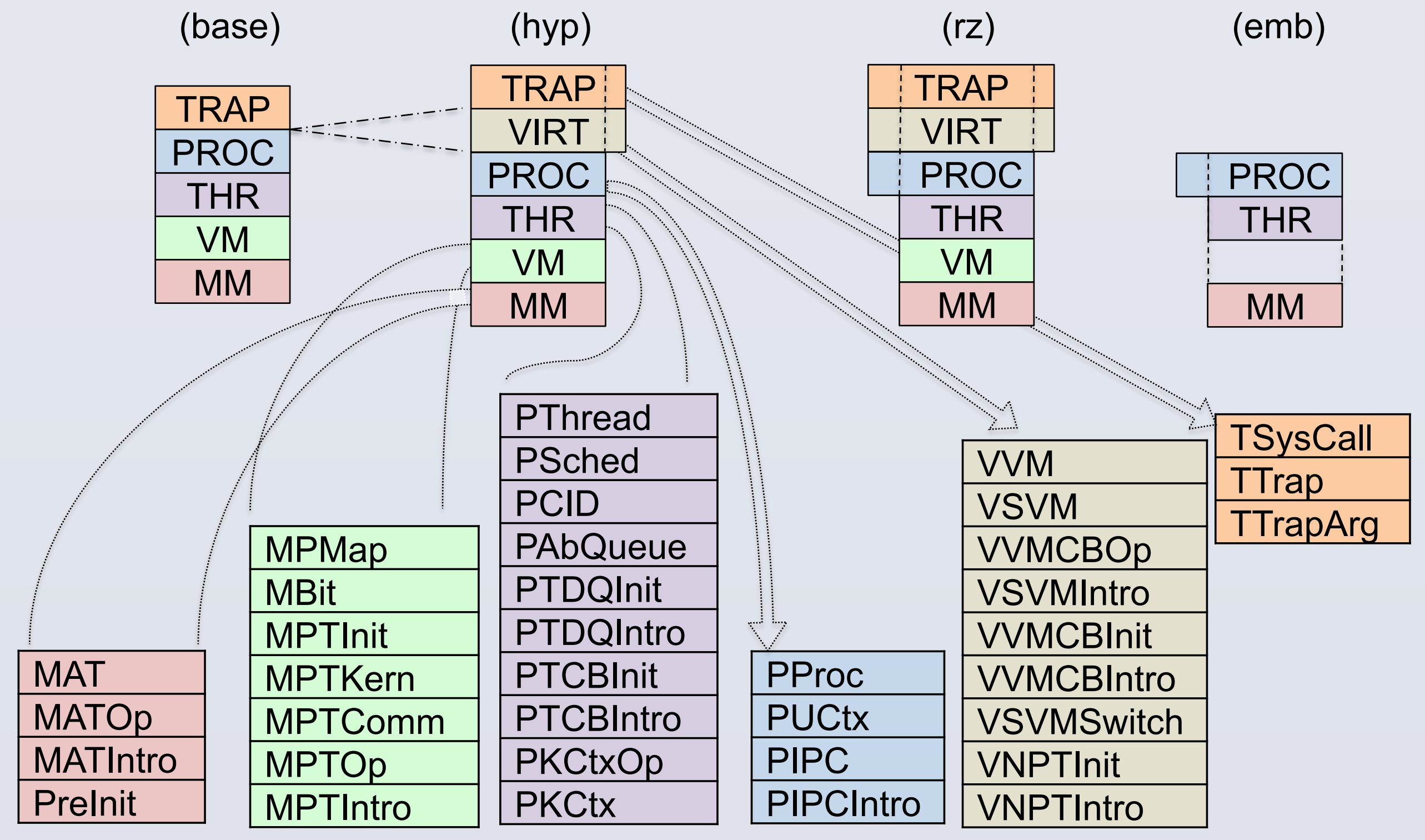
Programming & Compiling Layers



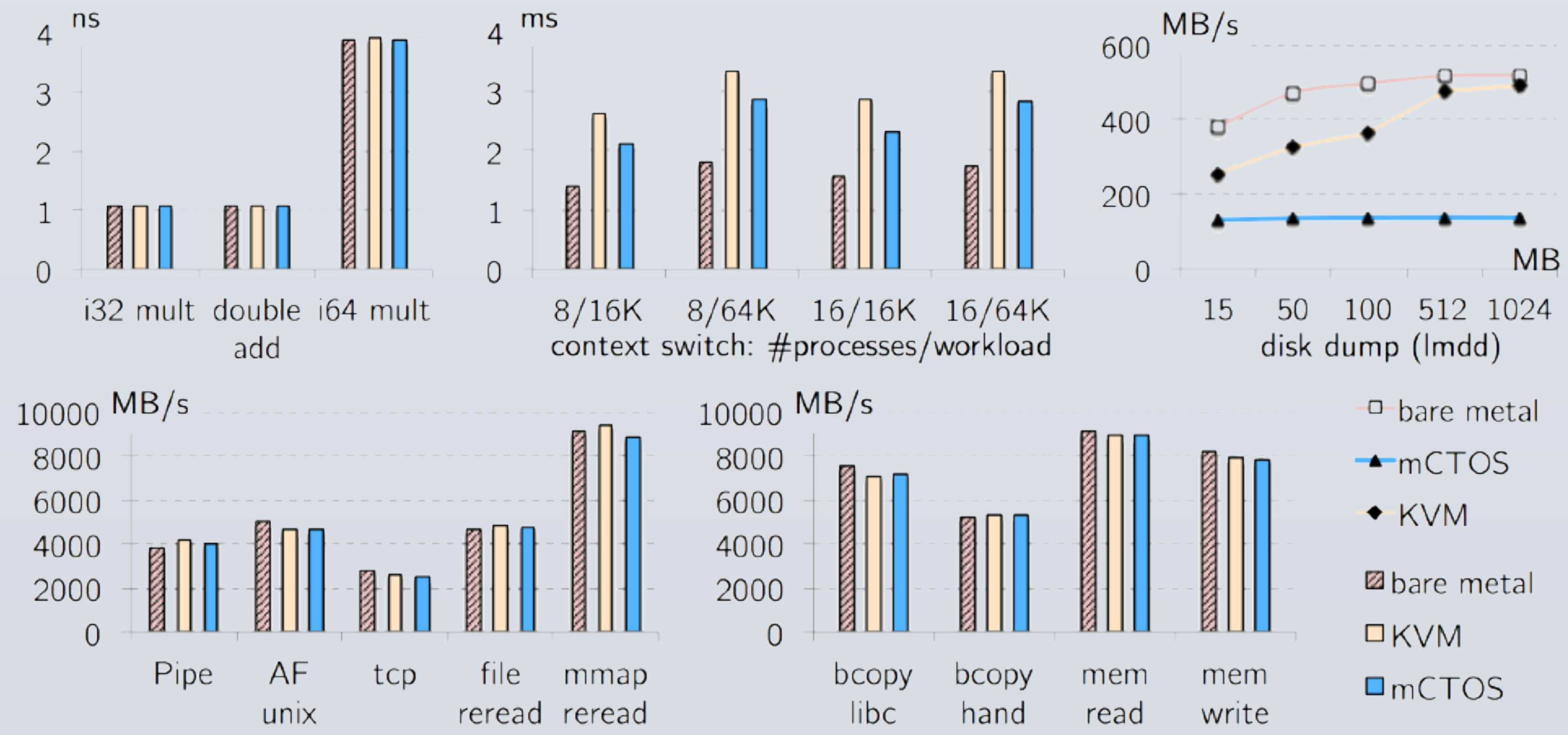
Variants of mCertiKOS Kernels

- Final theorem for mCertiKOS_hyp:

$$\forall \text{P}, 【 P \oplus CompCertX(mCertiKOS\_hyp) 】 (PreInit) \leq 【 P 】 (TSysCall)$$



Performance


ns
i32 mult
double add
i64 mult
ms
8/16K
8/64K
16/16K
16/64K
context switch: #processes/workload
MB/s
MB
disk dump (lmdd)
MB/s
Pipe
AF unix
tcp
file reread
mmap reread
MB/s
bcopy libc
bcopy hand
mem read
mem write
bare metal
mCTOS
KVM
bare metal
KVM
mCTOS