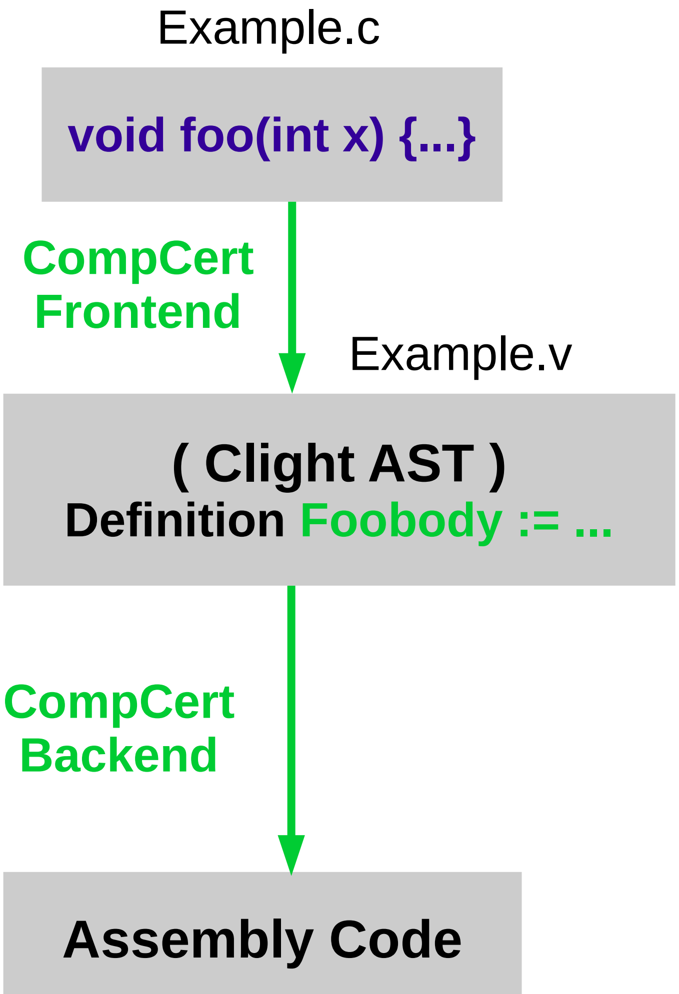
Verification using the Verified Software Toolchain

Lennart Beringer

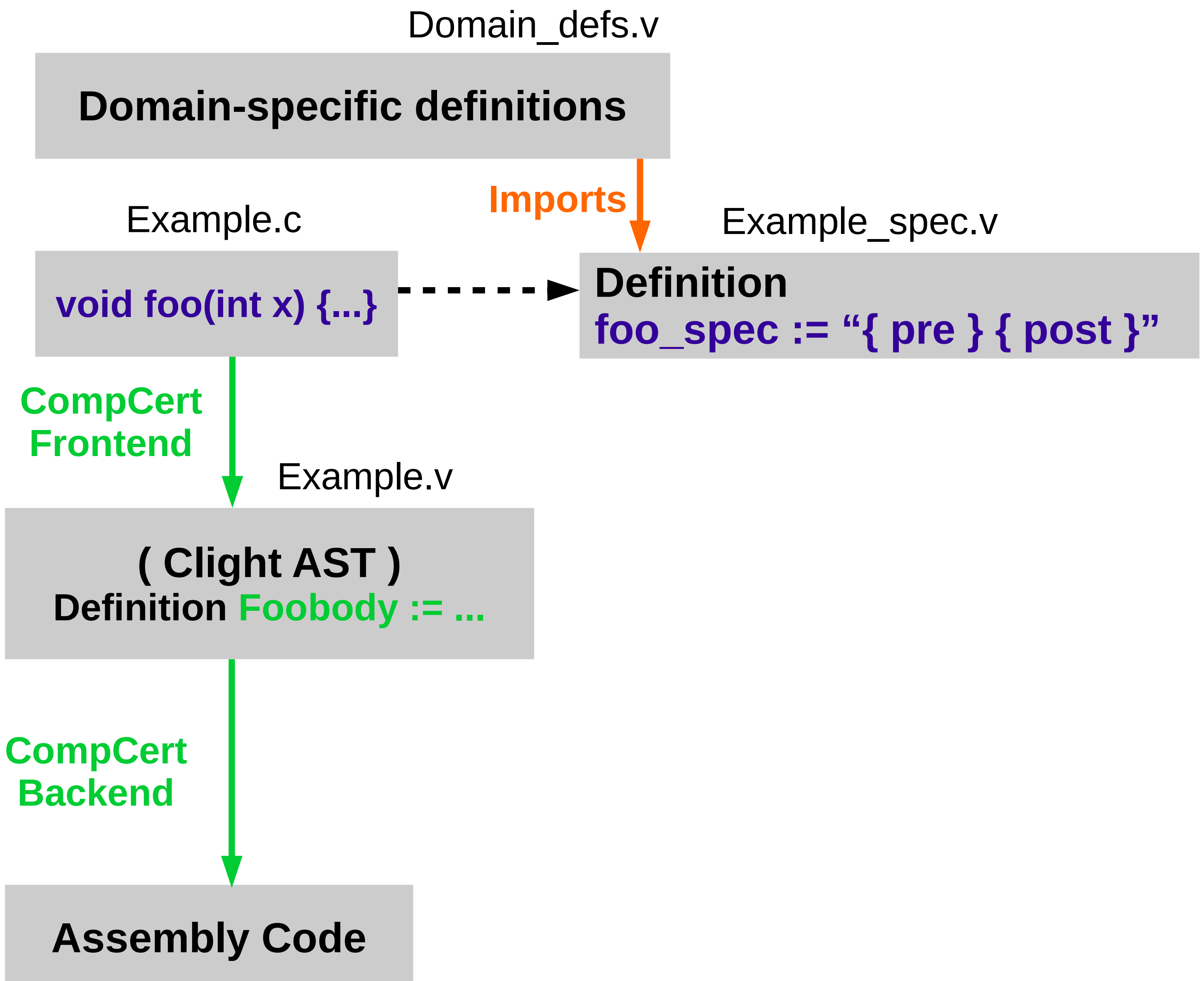
PRINCETON UNIVERSITY

CPS Workshop @ CMU, May 12, 2017

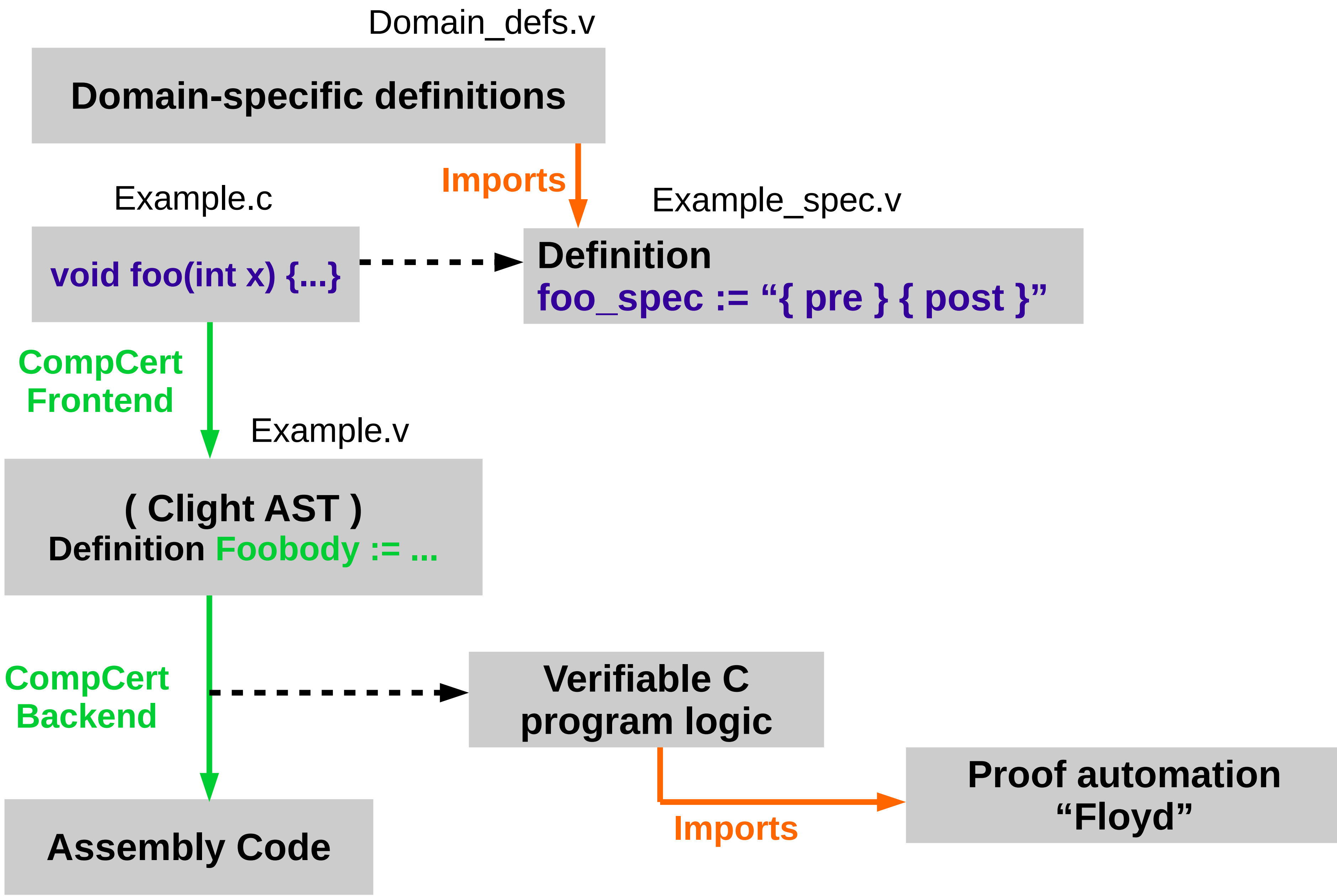
VST verification flow



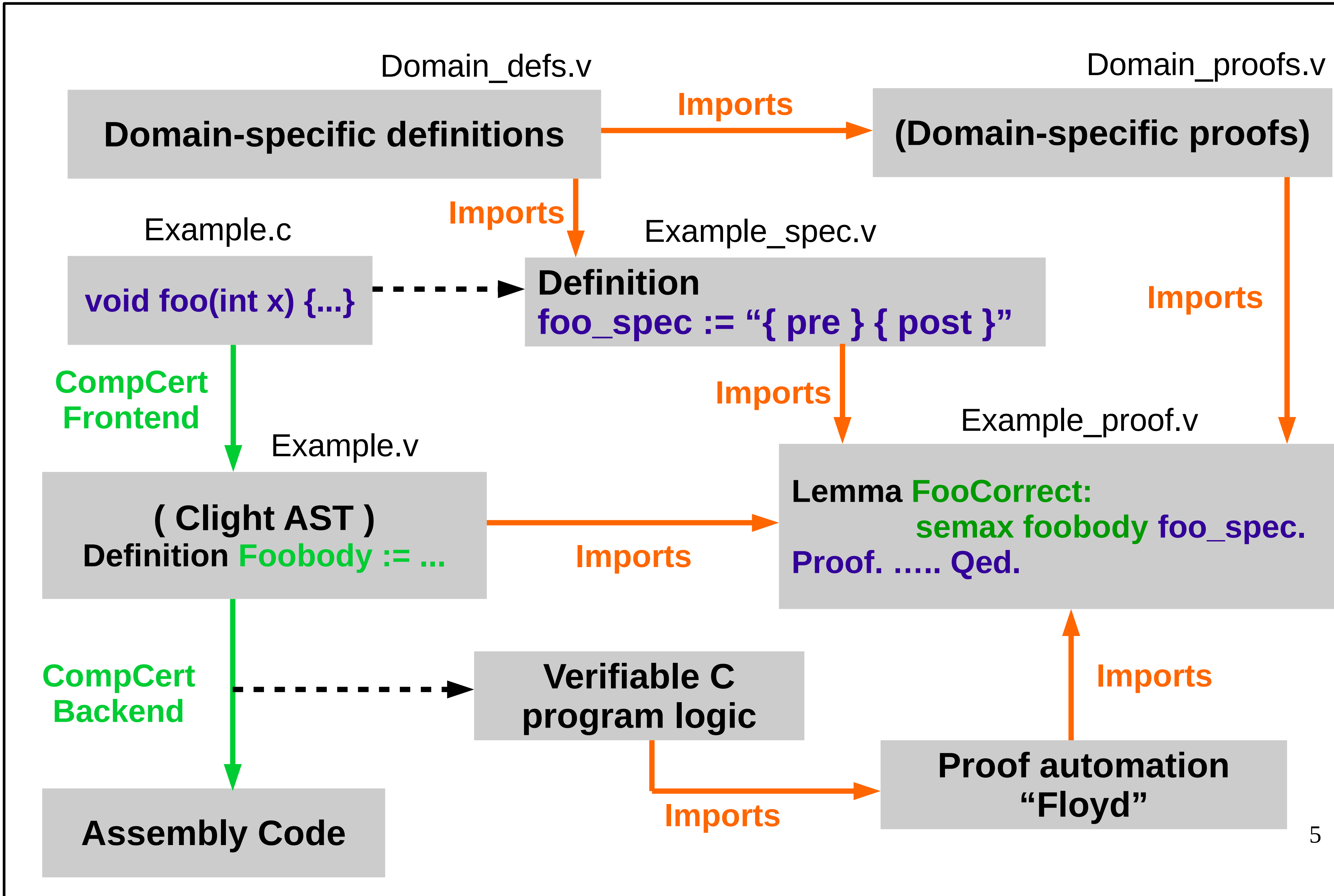
VST verification flow

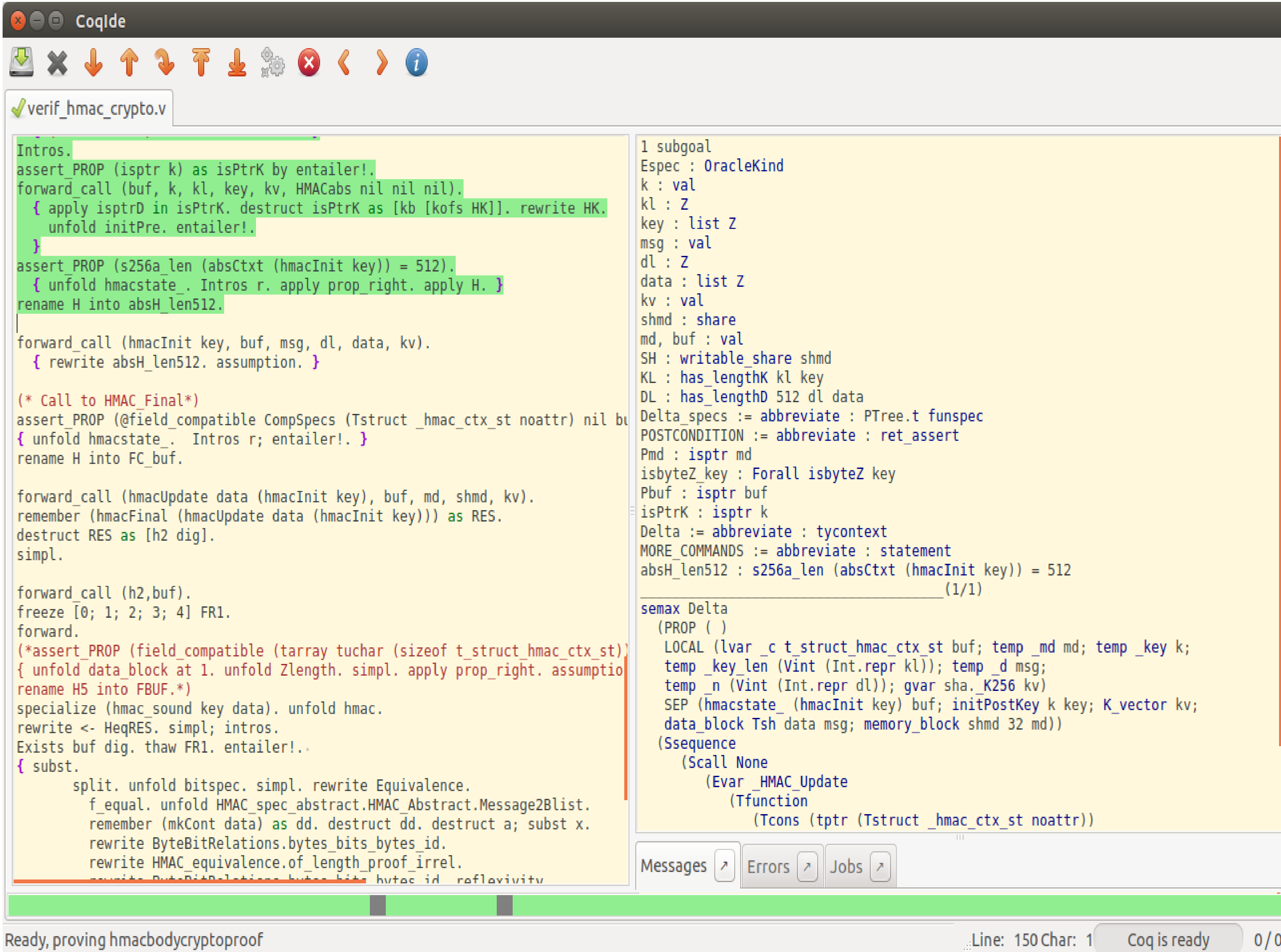


VST verification flow



VST verification flow



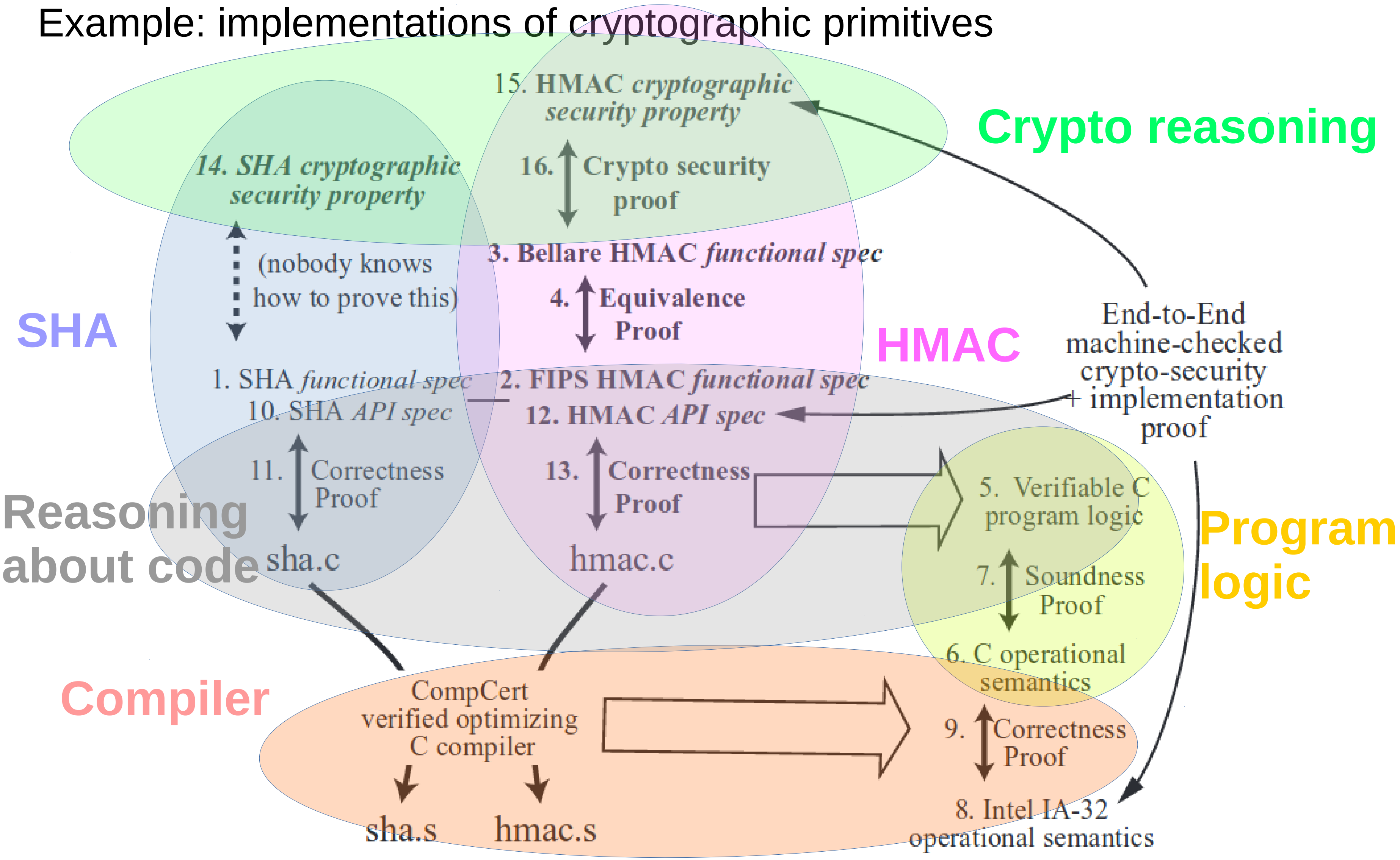

CoqIde
verif_hmac_crypto.v
Intros.
assert_PROP (isptr k) as isPtrK by entailer!.
forward_call (buf, k, kl, key, kv, HMACabs nil nil nil).
{ apply isptrD in isPtrK. destruct isPtrK as [kb [kofs HK]]. rewrite HK.
unfold initPre. entailer!.
}
assert_PROP (s256a_len (absCtxt (hmacInit key)) = 512).
{ unfold hmacstate_. Intros r. apply prop_right. apply H. }
rename H into absH_len512.
forward_call (hmacInit key, buf, msg, dl, data, kv).
{ rewrite absH_len512. assumption. }
(* Call to HMAC_Final*)
assert_PROP (@field_compatible CompSpecs (Tstruct _hmac_ctx_st noattr) nil bu
{ unfold hmacstate_. Intros r; entailer!. }
rename H into FC_buf.
forward_call (hmacUpdate data (hmacInit key), buf, md, shmd, kv).
remember (hmacFinal (hmacUpdate data (hmacInit key))) as RES.
destruct RES as [h2 dig].
simpl.
forward_call (h2,buf).
freeze [0; 1; 2; 3; 4] FR1.
forward.
(*assert_PROP (field_compatible (tarray tuchar (sizeof t_struct_hmac_ctx_st))
{ unfold data_block at 1. unfold Zlength. simpl. apply prop_right. assumptio
rename H5 into FBUF.*)
specialize (hmac_sound key data). unfold hmac.
rewrite <- HeqRES. simpl; intros.
Exists buf dig. thaw FR1. entailer!.
{ subst.
split. unfold bitspec. simpl. rewrite Equivalence.
f_equal. unfold HMAC_spec_abstract.HMAC_Abstract.Message2Blist.
remember (mkCont data) as dd. destruct dd. destruct a; subst x.
rewrite ByteBitRelations.bytes_bits_bytes_id.
rewrite HMAC_equivalence.of_length_proof_irrel.
1 subgoal
Espec : OracleKind
k : val
kl : Z
key : list Z
msg : val
dl : Z
data : list Z
kv : val
shmd : share
md, buf : val
SH : writable_share shmd
KL : has_lengthK kl key
DL : has_lengthD 512 dl data
Delta_specs := abbreviate : PTree.t funspec
POSTCONDITION := abbreviate : ret_assert
Pmd : isptr md
isbyteZ_key : Forall isbyteZ key
Pbuf : isptr buf
isPtrK : isptr k
Delta := abbreviate : tycontext
MORE_COMMANDS := abbreviate : statement
absH_len512 : s256a_len (absCtxt (hmacInit key)) = 512
(1/1)
semax Delta
(PROP ( )
LOCAL (lvar _c t_struct_hmac_ctx_st buf; temp _md md; temp _key k;
temp _key_len (Vint (Int.repr kl)); temp _d msg;
temp _n (Vint (Int.repr dl)); gvar sha._K256 kv)
SEP (hmacstate_ (hmacInit key) buf; initPostKey k key; K_vector kv;
data_block Tsh data msg; memory_block shmd 32 md))
(Ssequence
(Scall None
(Evar _HMAC_Update
(Tfunction
(Tcons (tptr (Tstruct _hmac_ctx_st noattr))
Messages
Errors
Jobs
Ready, proving hmacbodycryptoproof
Line: 150 Char: 1
Coq is ready
0 / 0

VST: key properties

- Program logic:
  - expressive concurrent separation logic
  - { **Prop** P **Local** Q **Sep** R } c { **Prop** P’ **Local** Q’ **Sep** R’ }
  - specifications: extractable / executable programs, or propositional
  - formal soundness proof in Coq w.r.t. Clight op.semantics
- Proof automation “Floyd”:
  - forward symbolic execution
  - domain specific tactics for entailments etc
- Guarantees:
  - safety: no undefined behavior, memory safety (buffer overrun,…), div-by-0,…
  - confinement: no read/writes outside specified memory region
- Major design goals: compositionality, support of “real C”

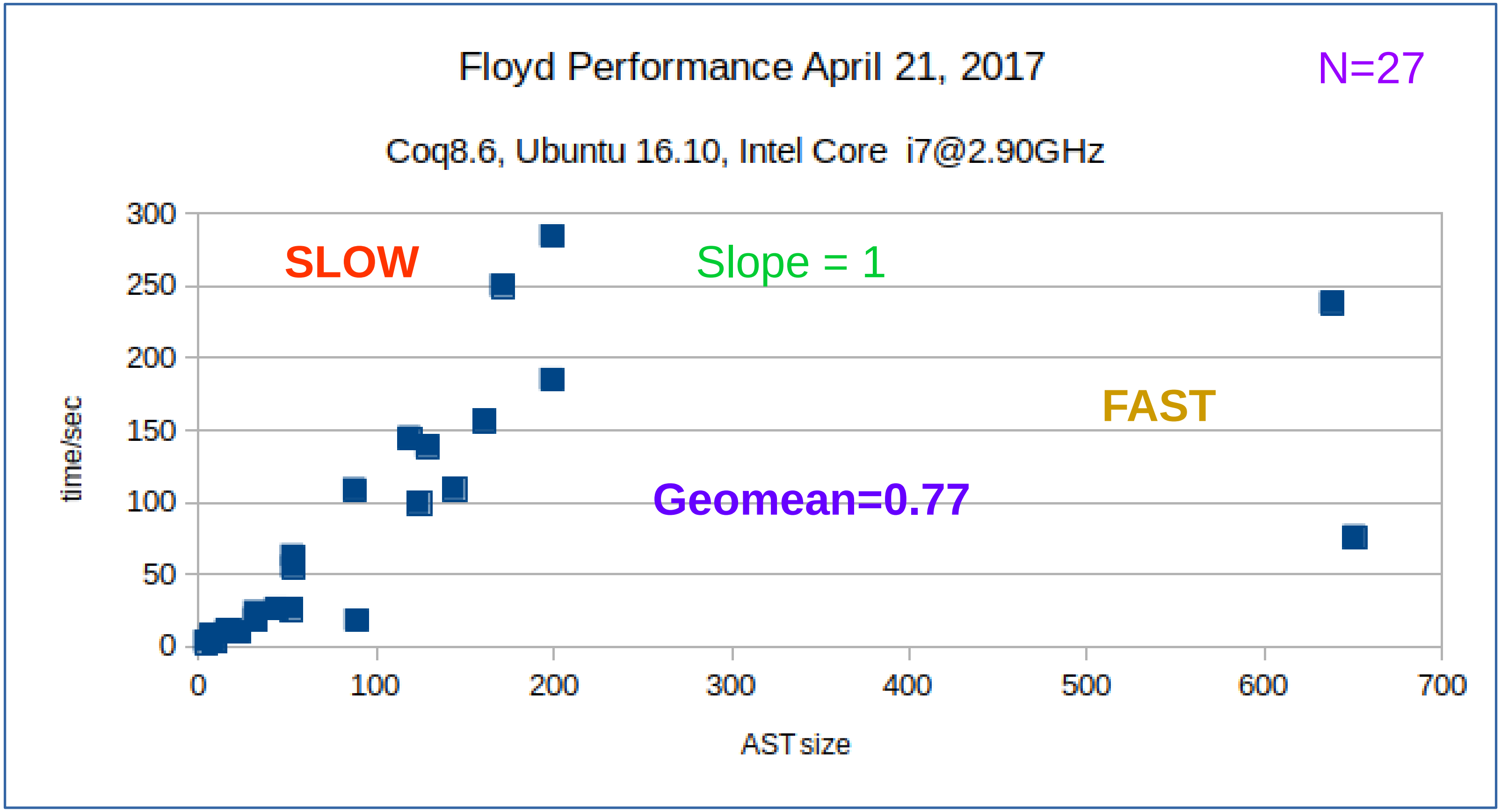
Compositional verification, end to end

- **horizontally**: compose verified code modules
- **vertically**: separate code verification from model-level reasoning
- Example: implementations of cryptographic primitives

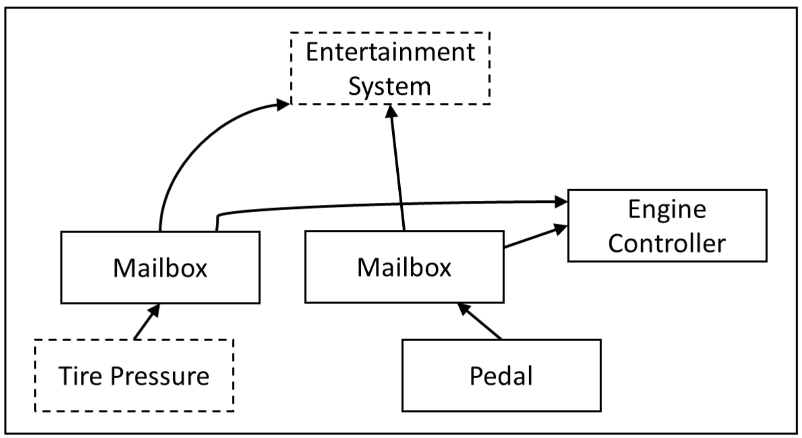


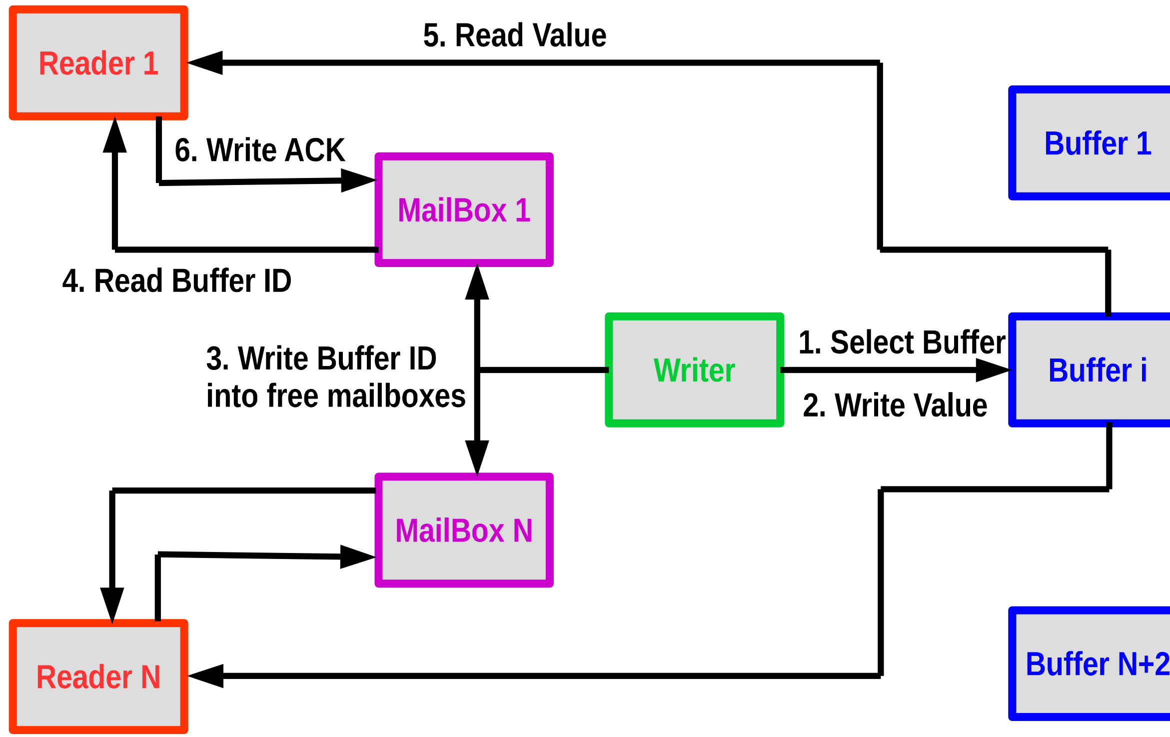
Automation & Performance

- assertions in **canonical form**: **PROP (P) LOCAL (Q) SEP (R)**
- SL proof rules for C complex! Many entailments!
- full employment theorem for tactics programmers
- horizontal frame, not vertical: **PROP (P) LOCAL (Q) SEP (R) FR (F)**



VST 4 CPS (1): communication protocols for sensor networks

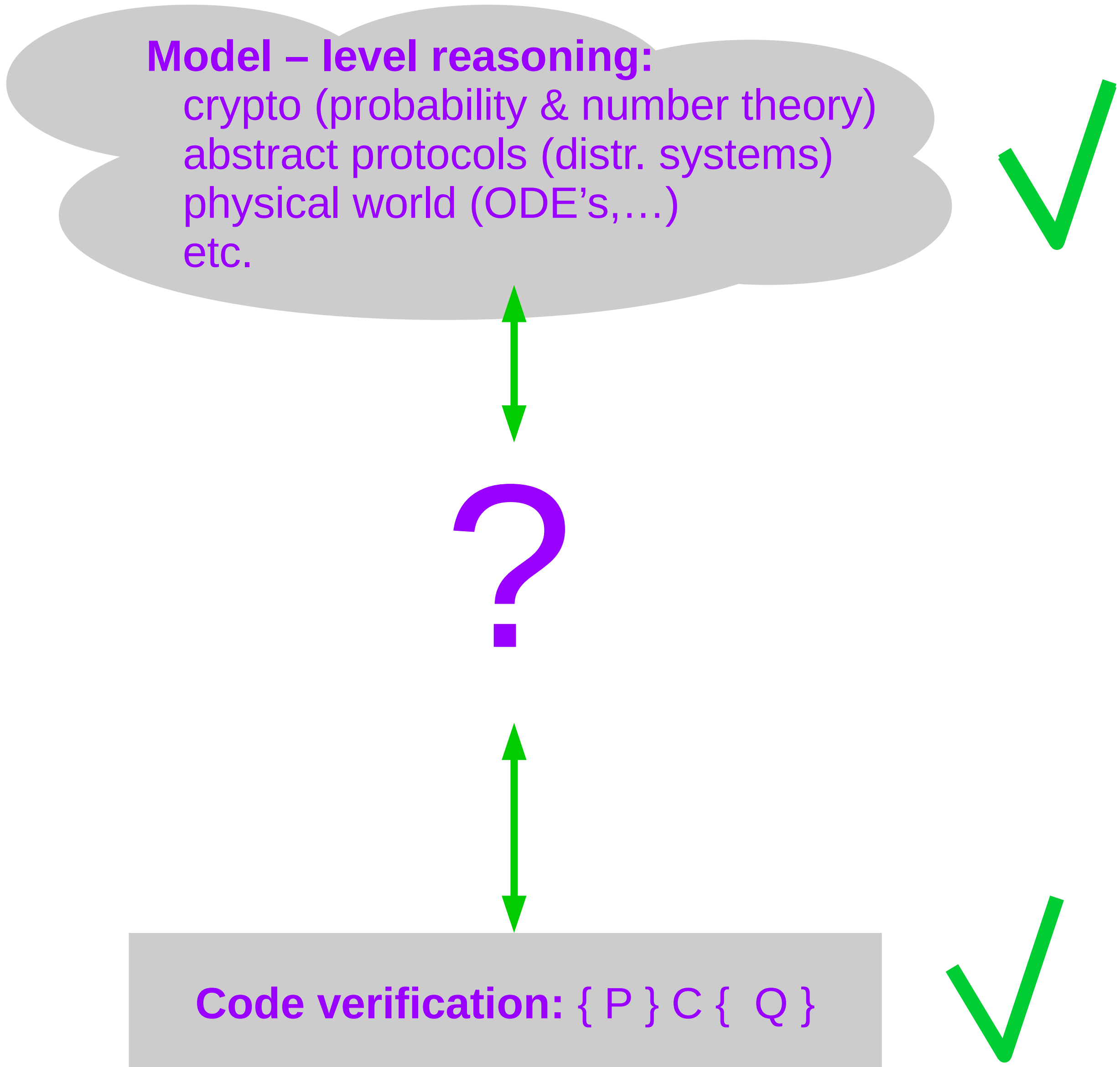




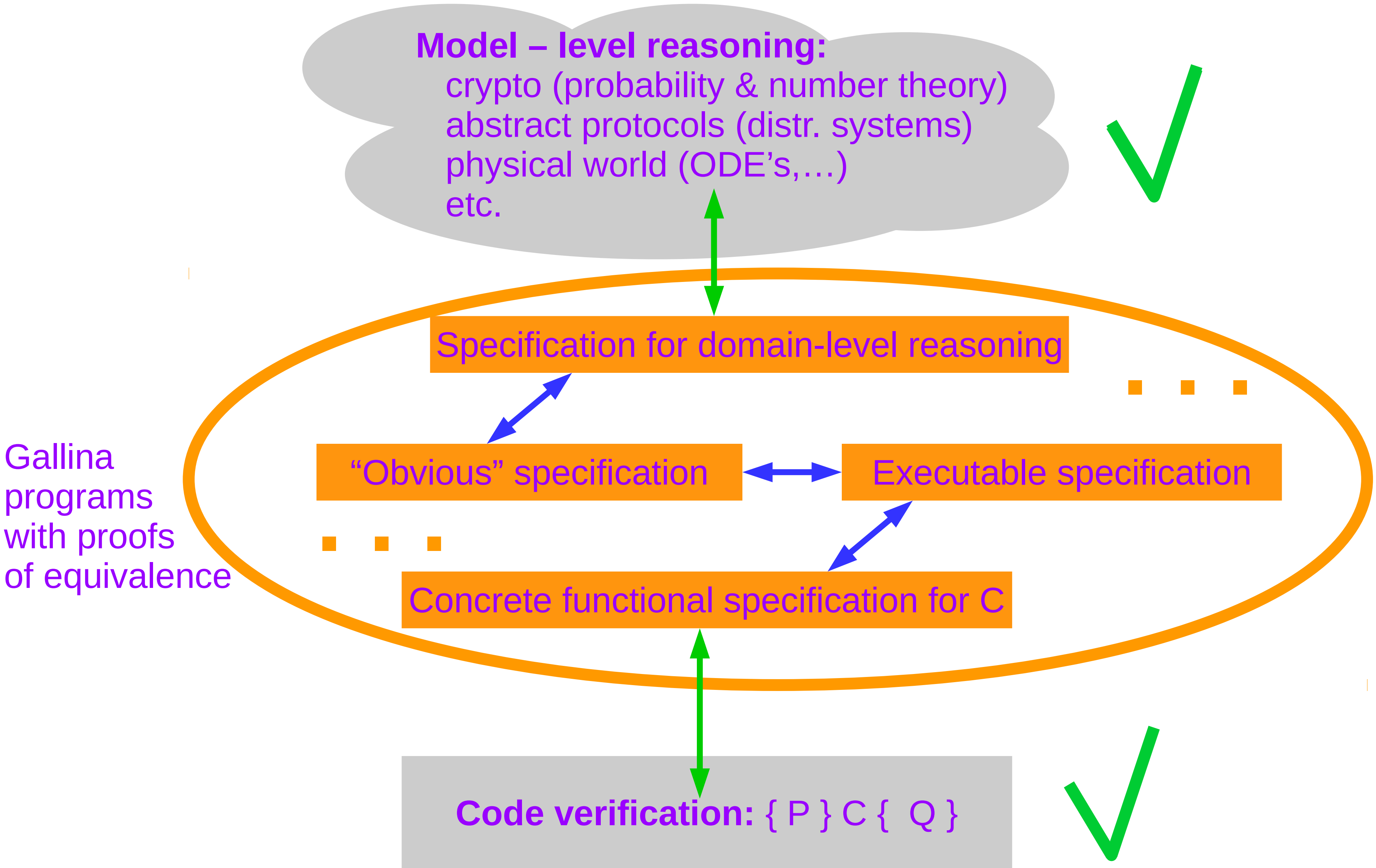
(Reader allowed to miss values. Existence of N+2 Buffers ensures non-blocking)

- CSL rules for lock-acquire / release:
  local thread gains / relinquishes resources
- also applicable to CAS, **atomic exchange**,...

VST 4 CPS (2):
Linking code verification & model-level reasoning

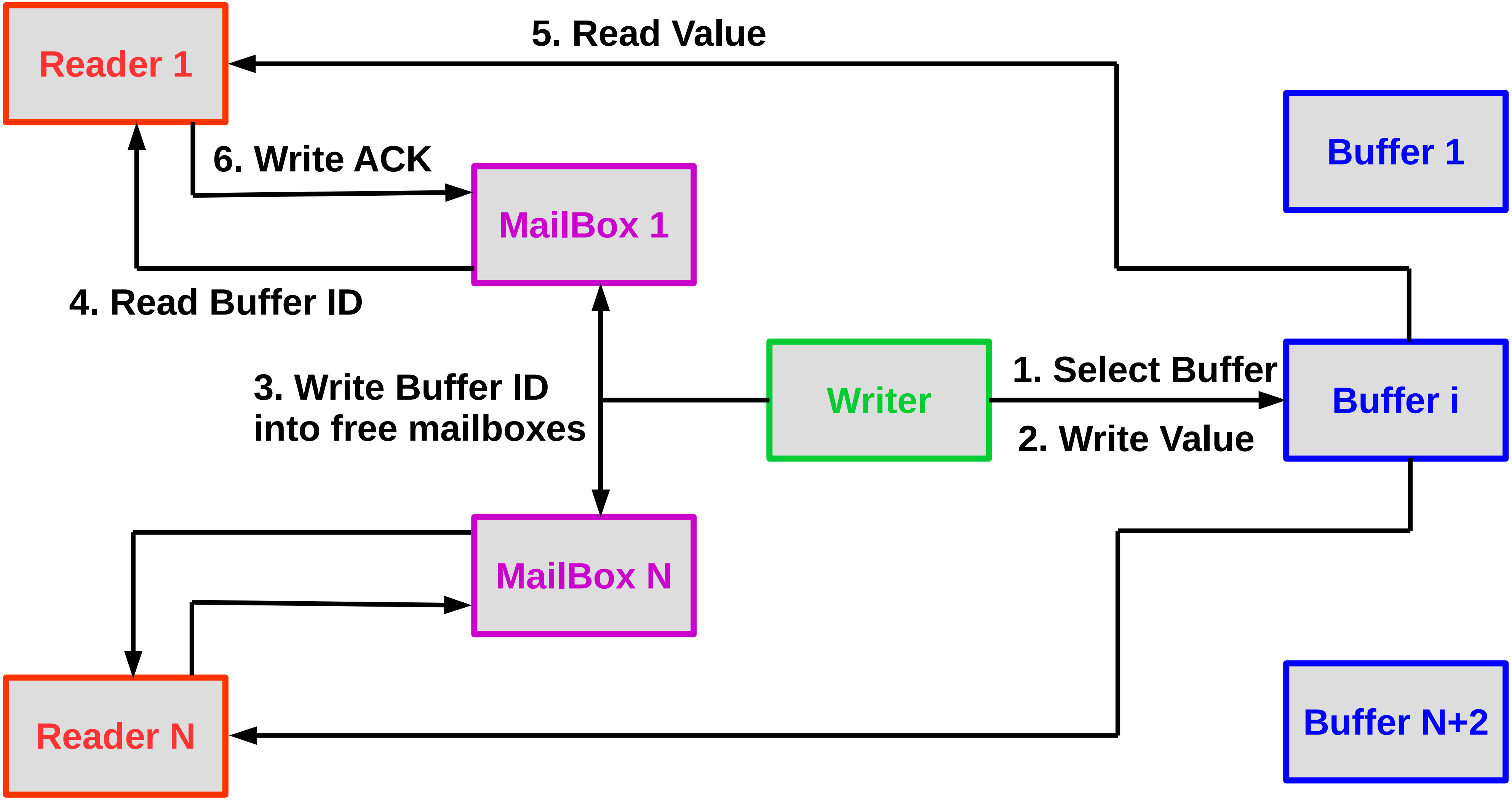


VST 4 CPS (2):
Linking code verification & model-level reasoning

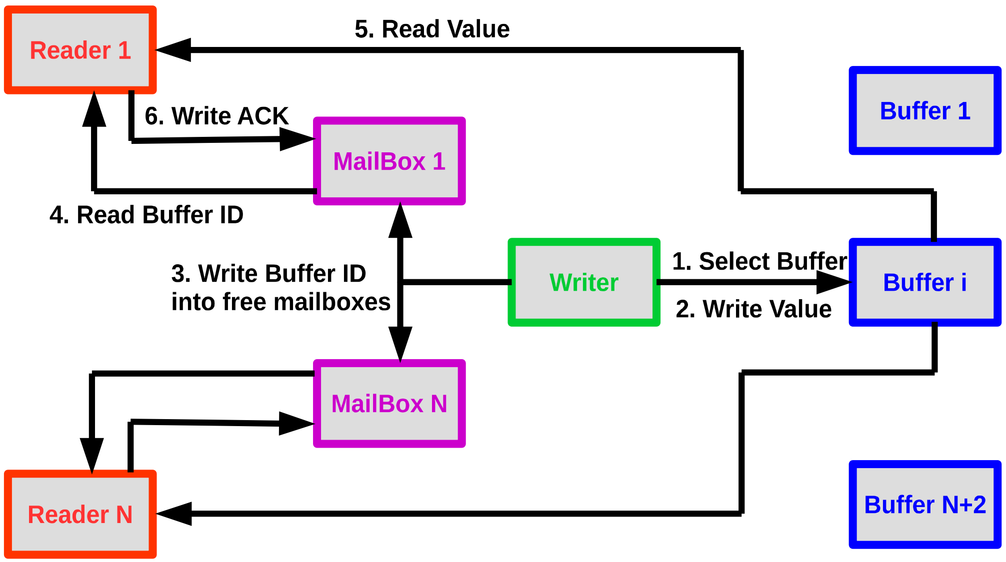


Vision

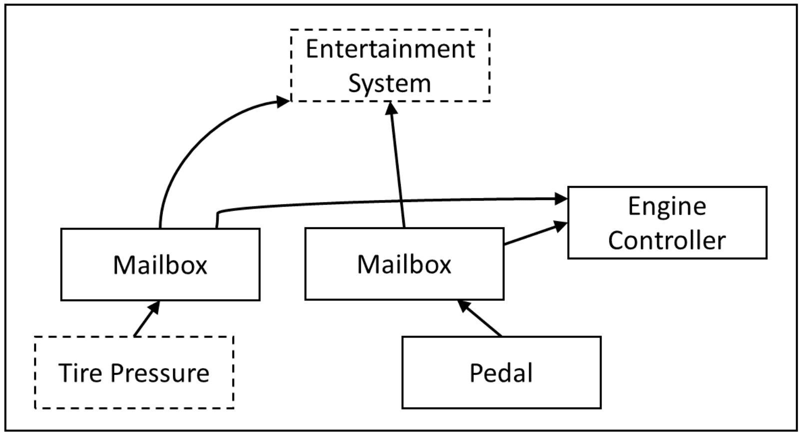
- Interactive proof assistant as “integrated development environment” for high-assurance software
- Other components: foundationally verified
  - hypervisors/OS-kernels
  - processor implementations
  - network stacks and other libraries
- Exemplary interface specifications under development in NSF Expedition



(Reader allowed to miss values. Existence of N+2 Buffers ensures non-blocking)



(Reader allowed to miss values. Existence of N+2 Buffers ensures non-blocking)



Entertainment System
Engine Controller
Mailbox
Mailbox
Tire Pressure
Pedal