Preserve safety by verifying only a small part of the system. Assure trust by protecting verified parts.

**Preserve safety**

- **Introduction**
  - Fielding new technologies is essential to preserve superiority. However, this is only possible if these technologies are validated for safety.

- **Challenges for validation**
  - Growing system complexity
  - Changing behavior at runtime (e.g., machine learning)
  - Interactions with physical world (e.g., vehicles)
    - Correct value
    - At right time (before crash)

- **Methods**
  - **Formal** automatic verification
    - Scalable
    - Unverified components
    - Monitored and enforced by verified components
    - Protected from unverified components
    - Verified from
      - Physics: verify reaction of physical model (e.g., physical vehicle)
      - Logic: correct value, with correct protection
      - Timing: At the right time
    - Verified protection

- **Results**
  - **Real-Time Mixed-Trust**
    - Computation
    - Verified protection mechanism (micro-hypervisor: UberXMHF)
    - Timing verification of combined trusted/untrusted (mixed-trust)
    - Physics verification of enforcement

**Verify PHYSICS**

Ensure that an unverified controller cannot violate safety bounds

**Lyapunov Function**

\[
V(x) = \begin{cases} 
0 & \text{if } x \in \mathcal{N}_s(x_{eq}) \\
\frac{1}{2} (x - x_{eq})^T (x - x_{eq}) & \text{otherwise}
\end{cases}
\]

- Correct value
- At right time (before crash)

**Verifying Timing**

Response time ≤ Deadline