

On dynamic **flow-sensitive** floating-label systems

Pablo Buiras¹, Deian Stefan²,
and Alejandro Russo¹

CHALMERS¹

STANFORD²
UNIVERSITY

COINS PhD student seminar
Tromsø, October 2014

Confidentiality

Security lattice

H



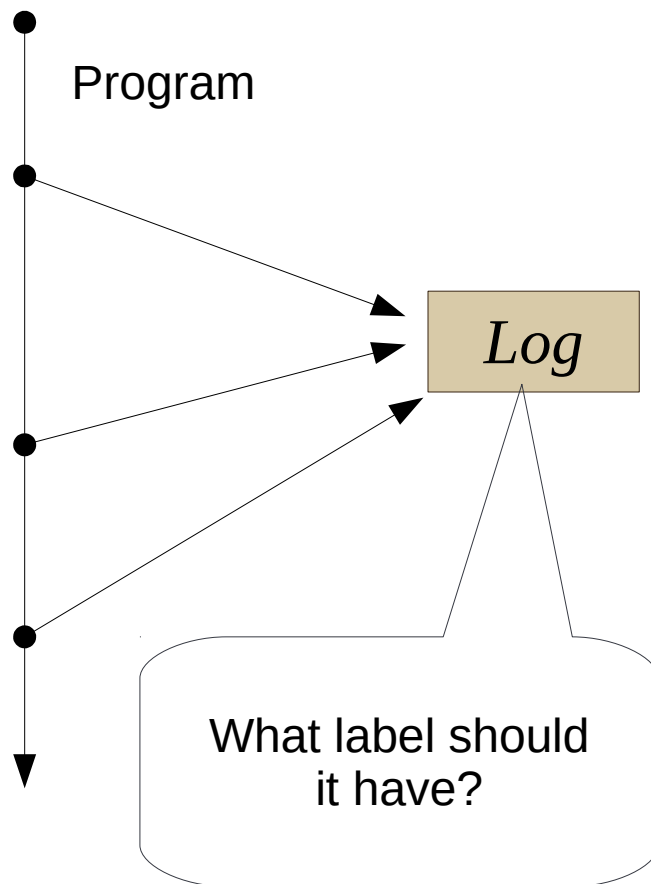
L

Arrows indicate **allowed** flows

$$L \preceq L \quad H \preceq H$$

$$L \preceq H$$

Logging and IFC



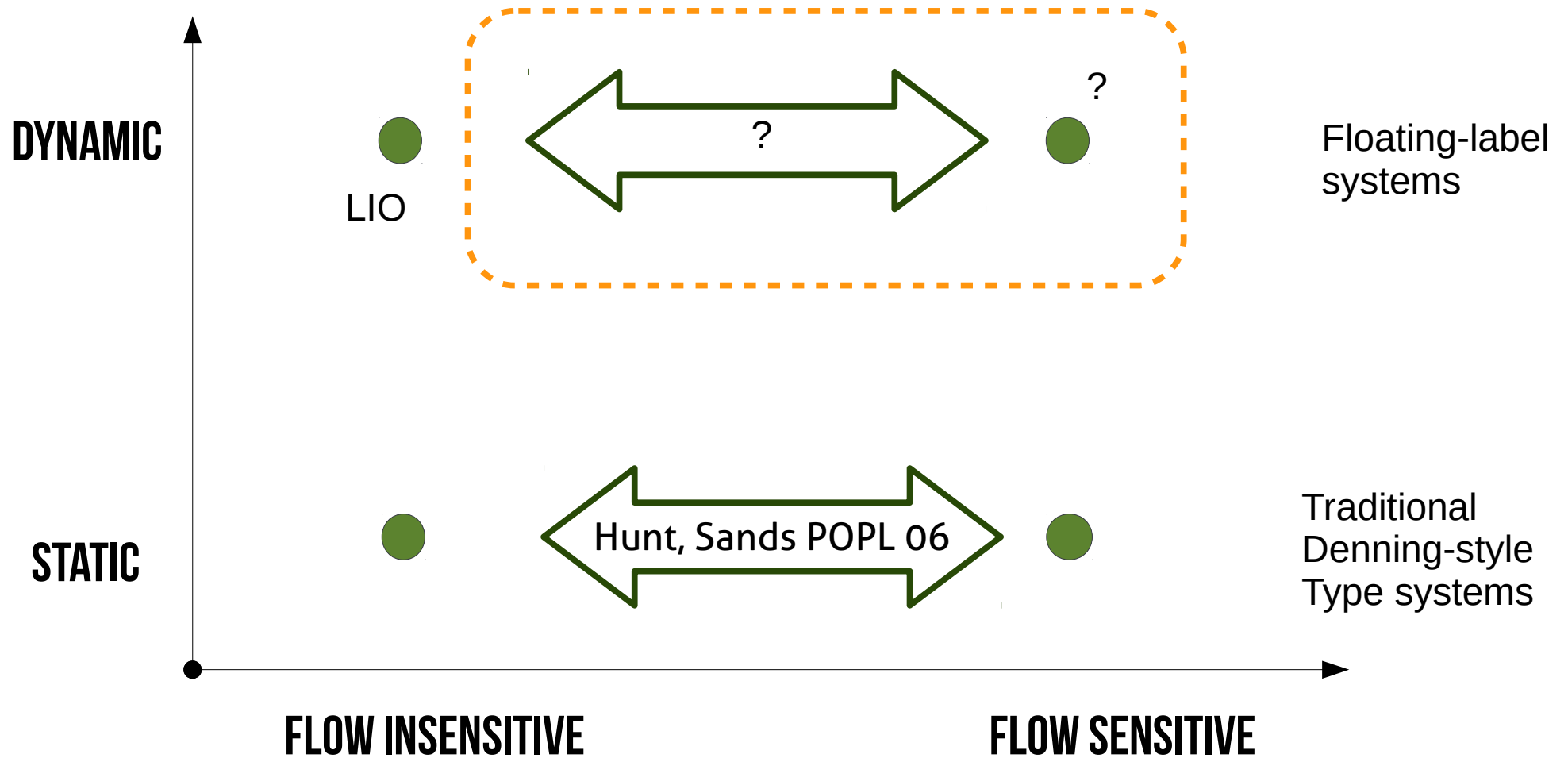
Flow-insensitive:

Fixed mapping from memory locations to security labels

Flow-sensitive:

Labels can change while the program runs

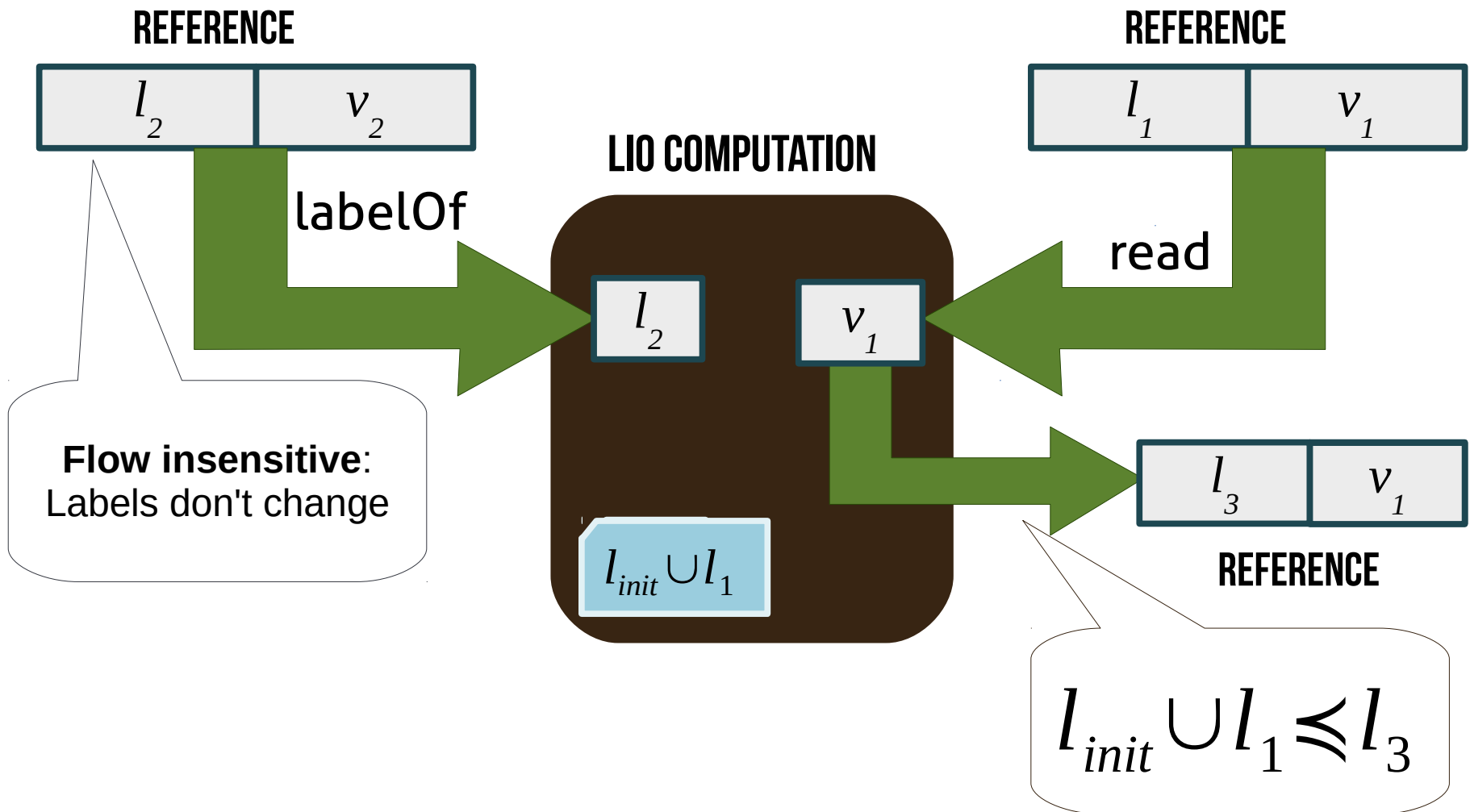
Our contributions



Flow-**i**nsensitive LIO

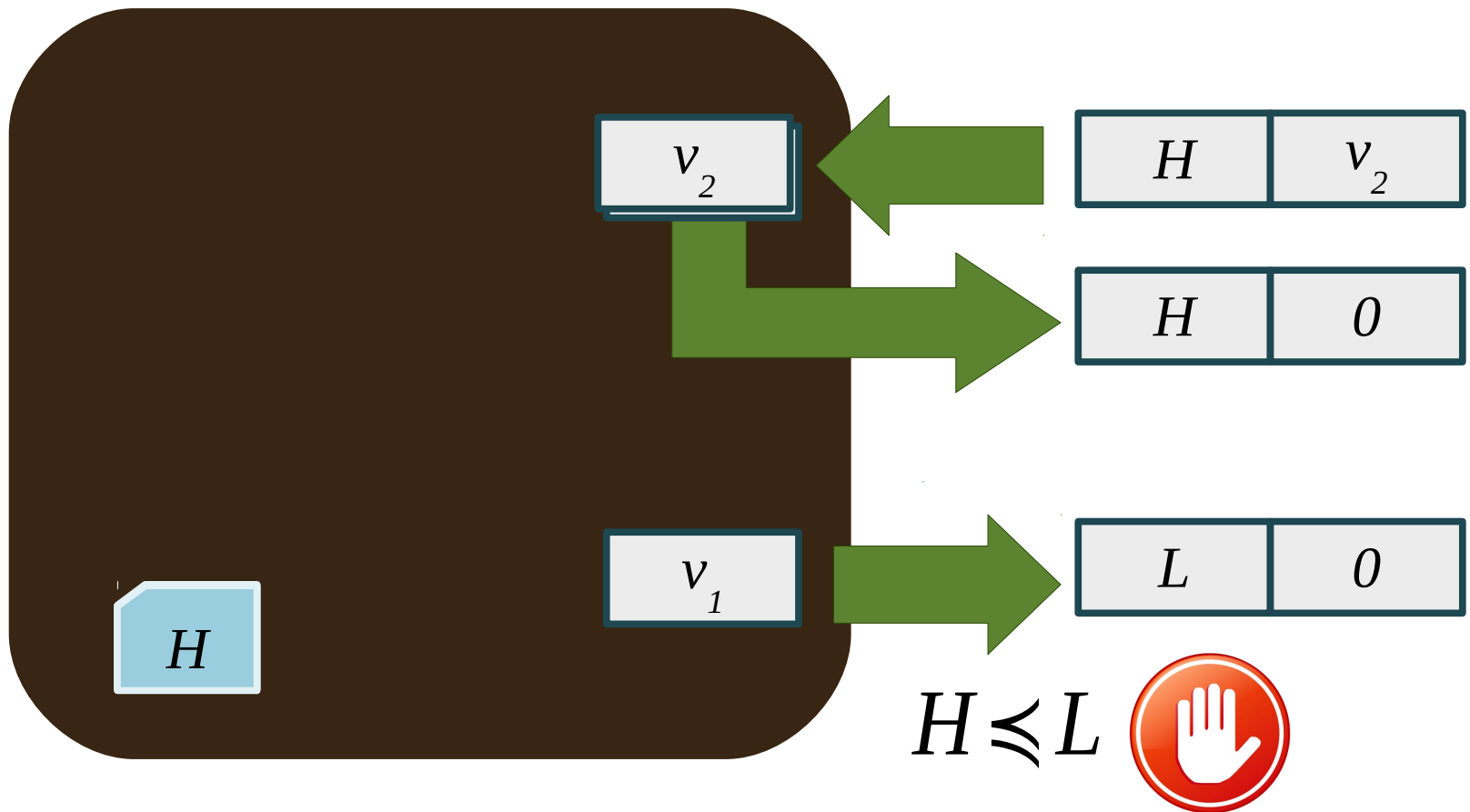


Floating label



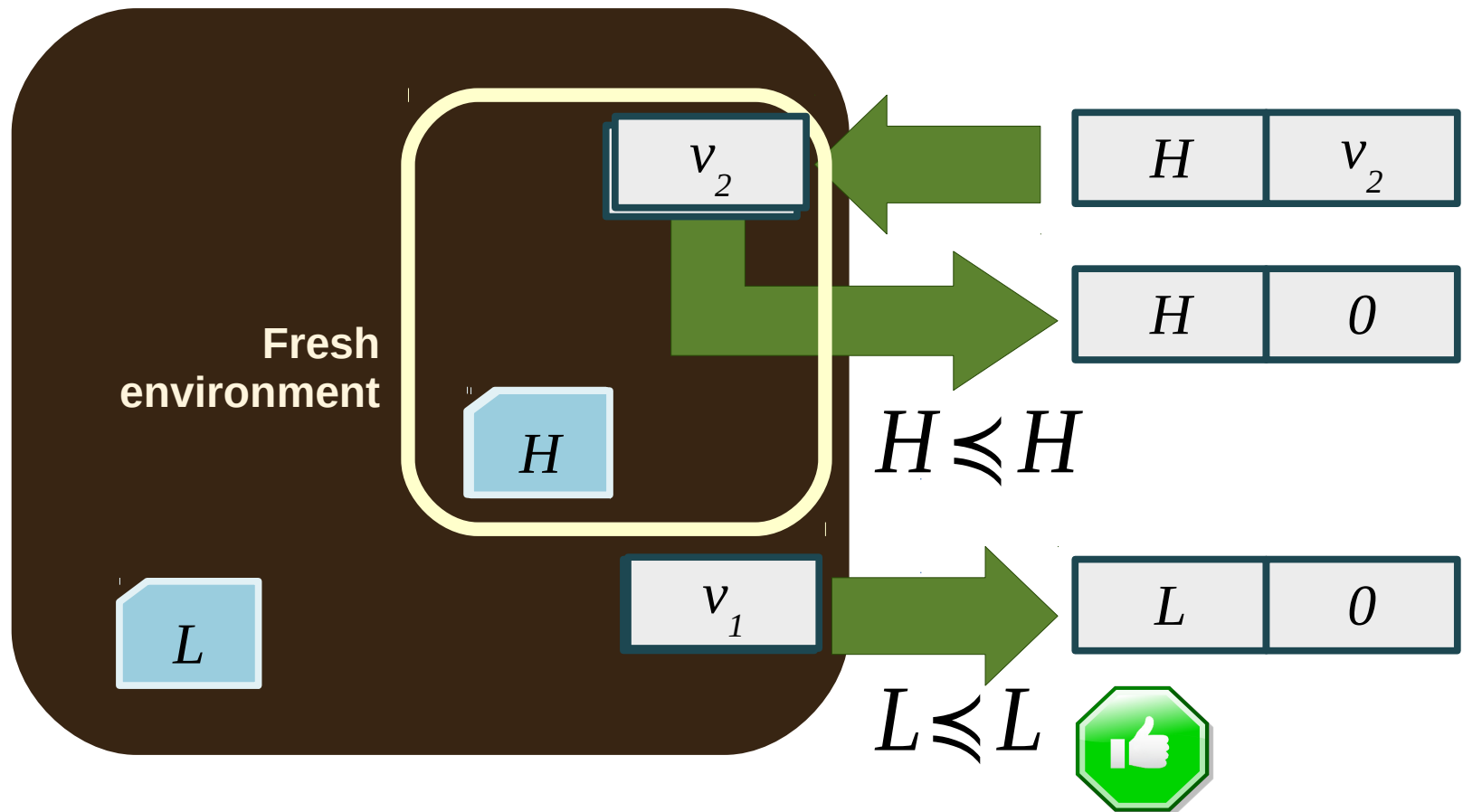
Label creep

LIO COMPUTATION



Label creep (2)

LIO COMPUTATION

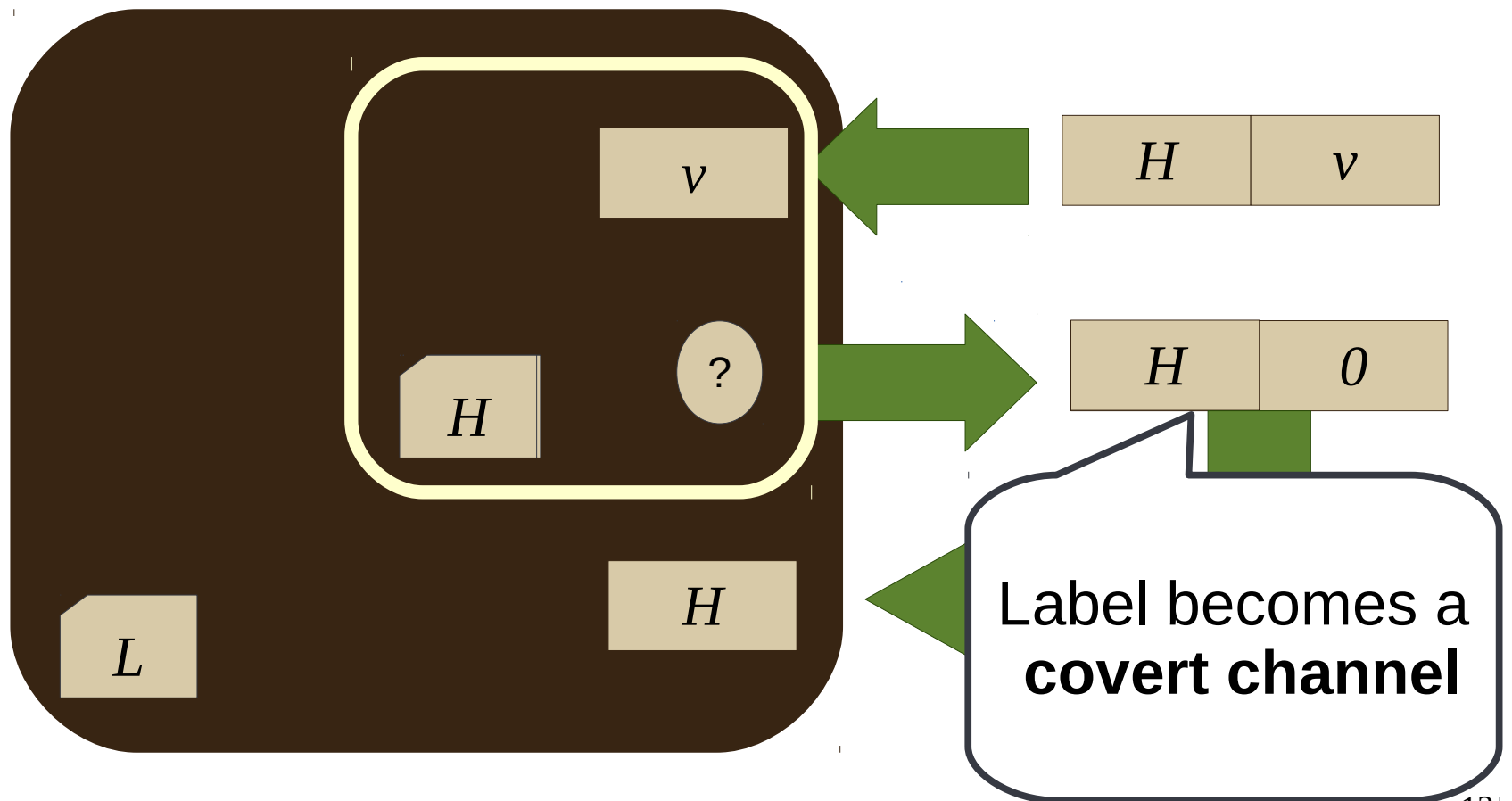


Flow-sensitive LIO



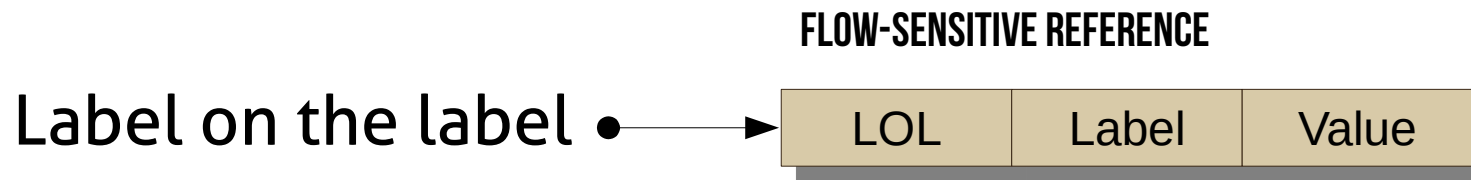
Naïve flow-sensitivity

- Change label when writing to the reference



Label on the label

- We must protect the label with another label!

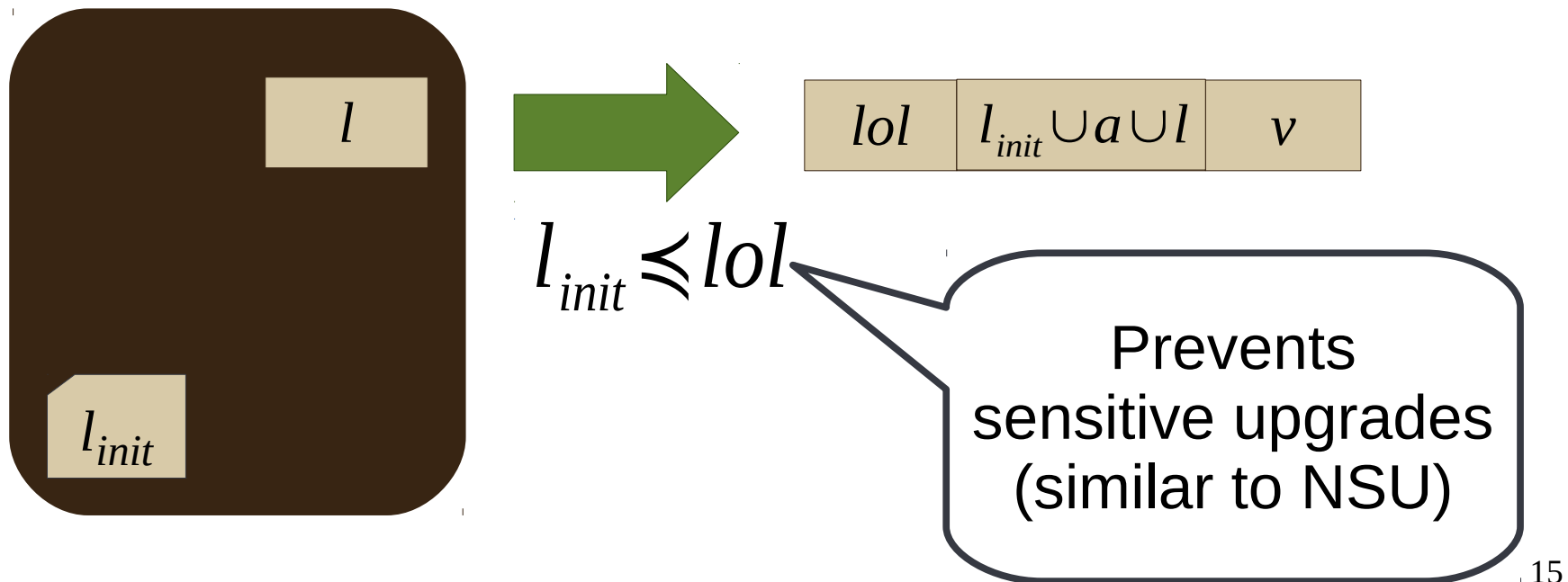


Invariant: $LOL \preceq Label$

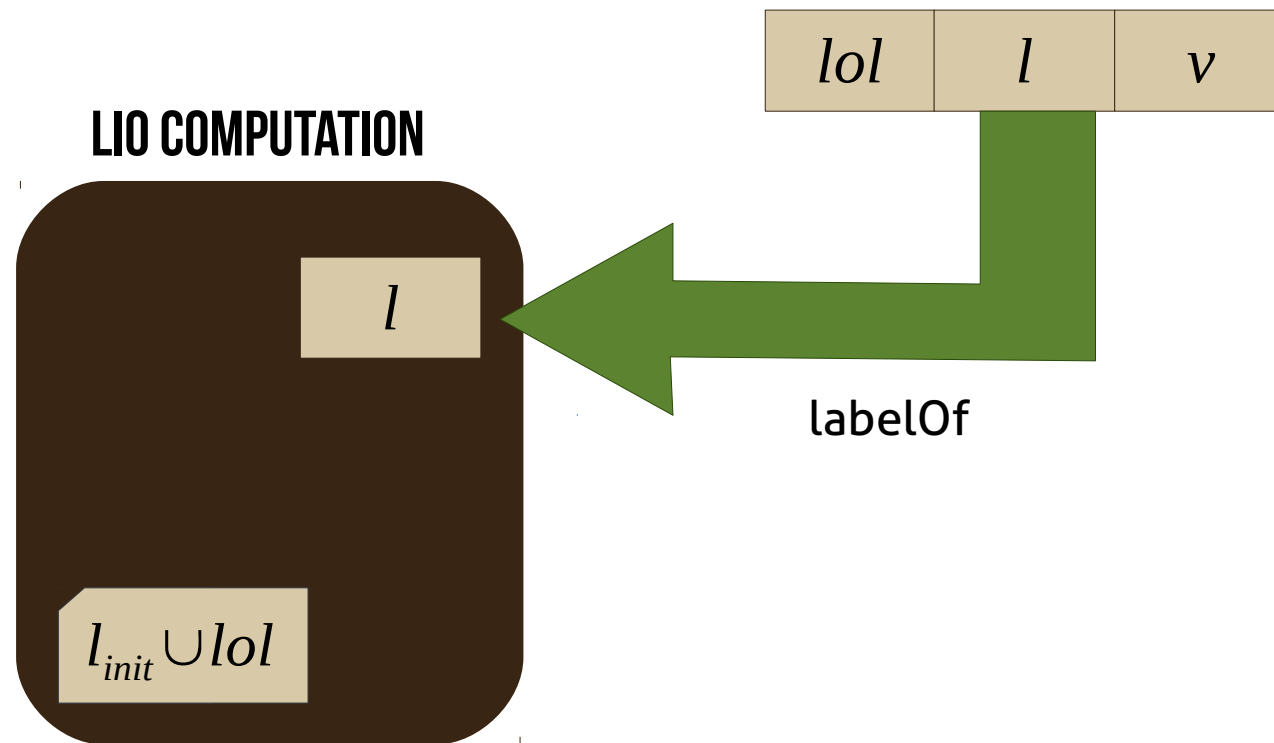
- LOL = current label at creation time

Upgrade

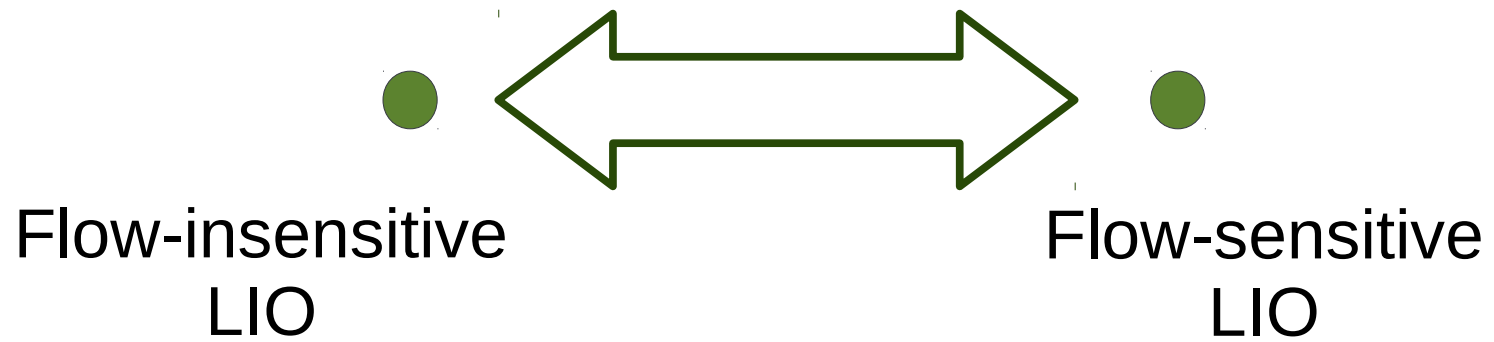
- Upgrades must be done **before** raising the current label
- Explicit **upgrade** operation (as in e.g. [Hedin, Sabelfeld SAC 14])



labelOf

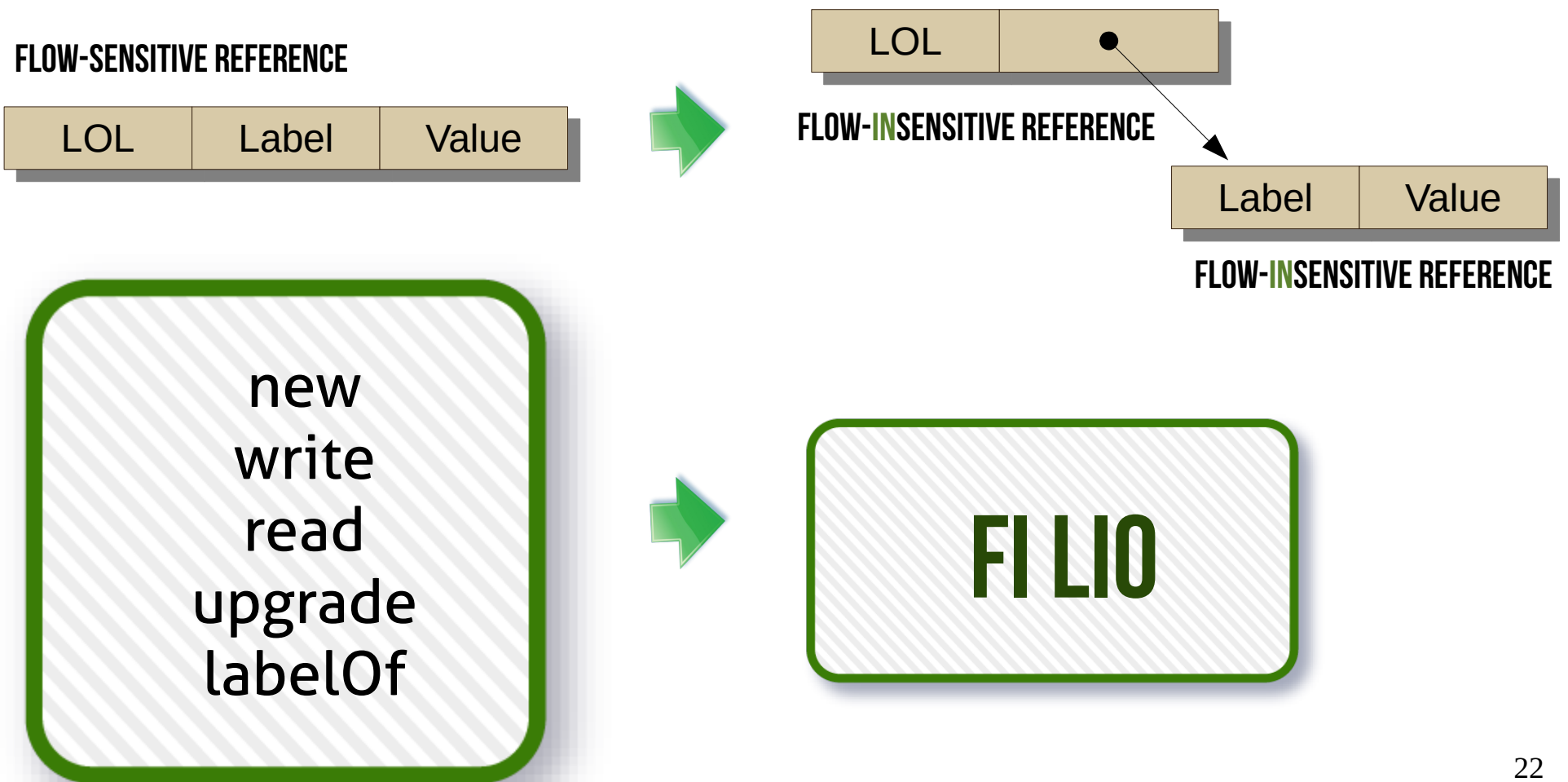


Embedding

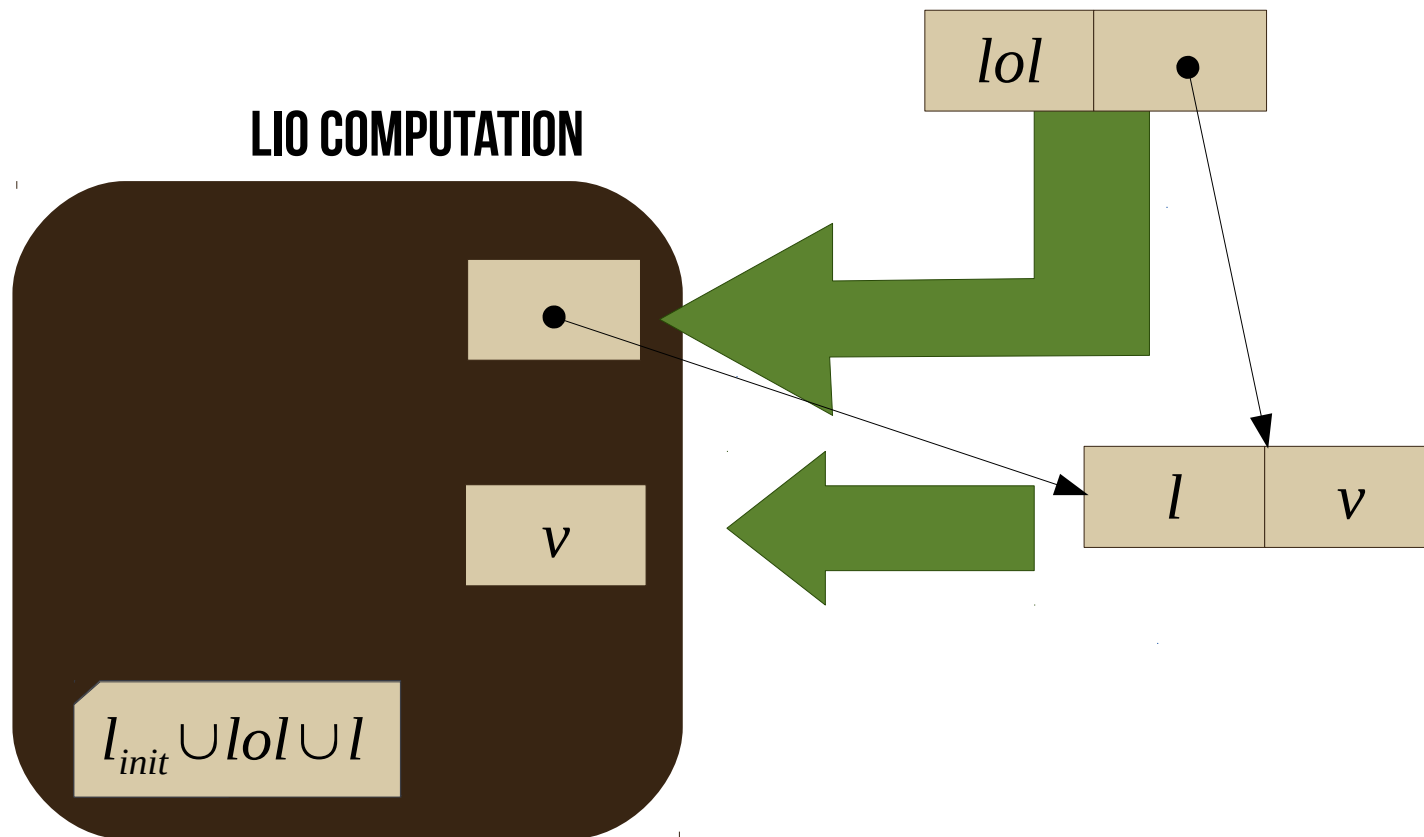


Embedding

- We can do this in the FI fragment of LIO!



Embedding of read



What we got

LIO + FS ✓

FI AND FS REFERENCES IN ONE SYSTEM



CONCURRENCY



SOUNDNESS FROM EMBEDDING

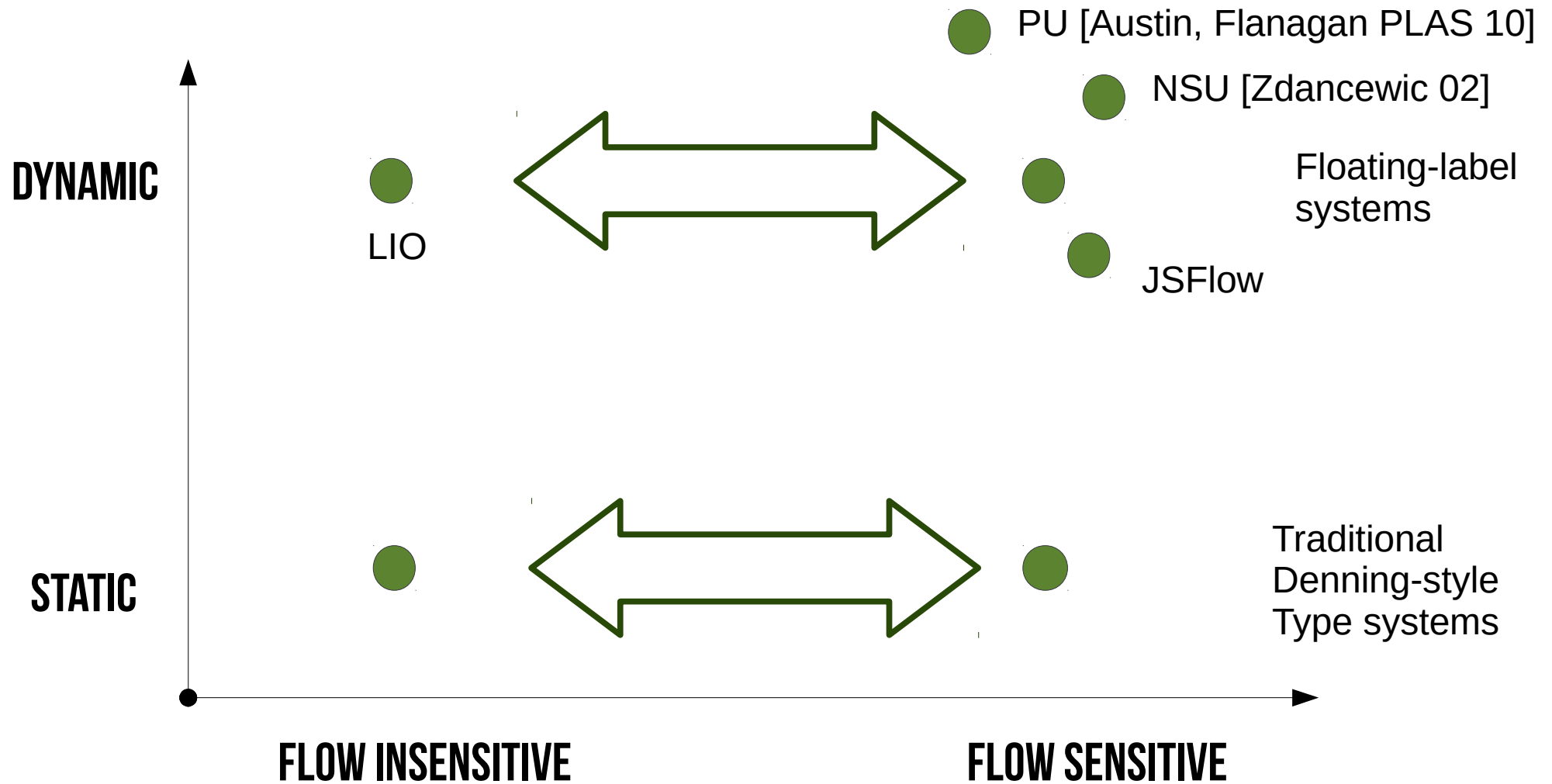


What we learnt

Flow sensitivity is tricky **but**

- First-class FI labelled references
 - Fresh environments
- } flow sensitivity

Related Work



Comparison with Related Work

- No-sensitive-upgrades [Zdancewic 02] can be encoded in a flow-insensitive enforcement
- Hard to compare with permissive-upgrades [Austin, Flanagan PLAS 2010]
- *Label on the label*
 - Isomorphic to *existence security labels* [Hedin et al CSF 2012] [Rafnsson, Sabelfeld CSF 2013]