



Ada, or How to Enforce Safety Rules at Compile Time

Jean-Pierre Rosen
Adalog
www.adalog.fr





Safety Integrity Levels and Segregation

- Railway systems: EN-50128 defines 5 “integrity levels”
 - From SIL0 (not critical) to SIL4 (highest criticality)
 - Constraints (and costs!) increase with SIL level
- Mixed criticality systems:
 - Same computer running various criticality applications
 - Same application with various criticality components
- How to make sure that unsafe components do not alter safe ones?
 - Validate all components at highest level (expensive!)
 - Hardware protection
 - Proofs





Segregation Requirements



- Components based architecture with only two levels: SIL0 (not certified) and SIL4 (certified) components
- Data
 - Data can be passed from SIL0 to SIL4
 - Deemed unreliable, SIL4 access must go through special gateways to check validity
 - No direct access of SIL4 data by SIL0 components
- Components
 - Some components are not by themselves SIL4, but may be called by SIL0 as well as SIL4 components
 - Classified as SIL4
 - SIL0 components shall not call other SIL4 components
 - SIL4 components shall call SIL0 components only through special isolation components



Child Unit and Visibility

- A package can be a *child* of another package (the *parent*)
 - Public child
 - **package** Parent.Child is ...
 - Private child
 - **private package** Parent.Child is ...
- A public child can be used by outer components
 - But it has no visible access to the parent's private part
- A private child can be used only by its parent and siblings (subsystem rooted at the parent)
 - But it has visibility on the parent's private part

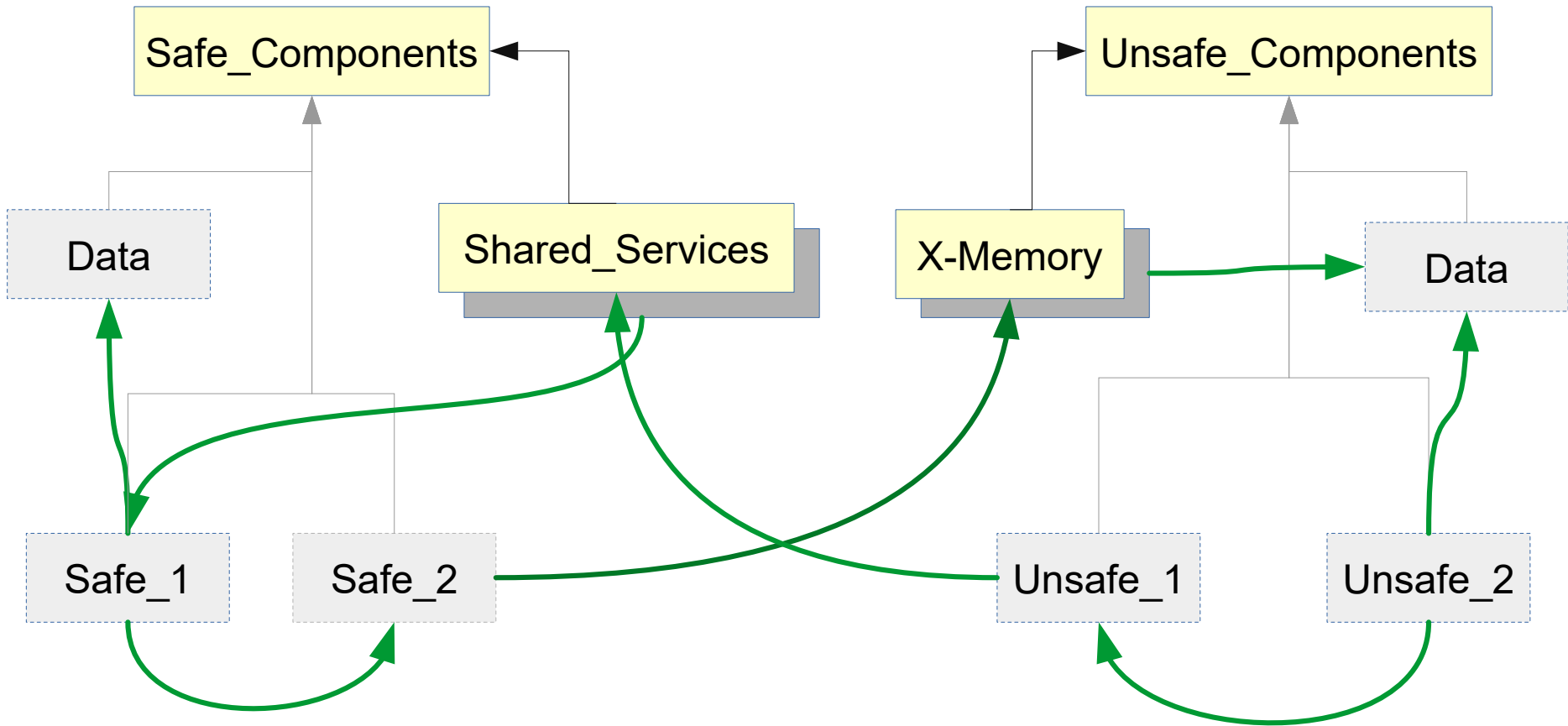


Structure



Public
unit/child

Private child



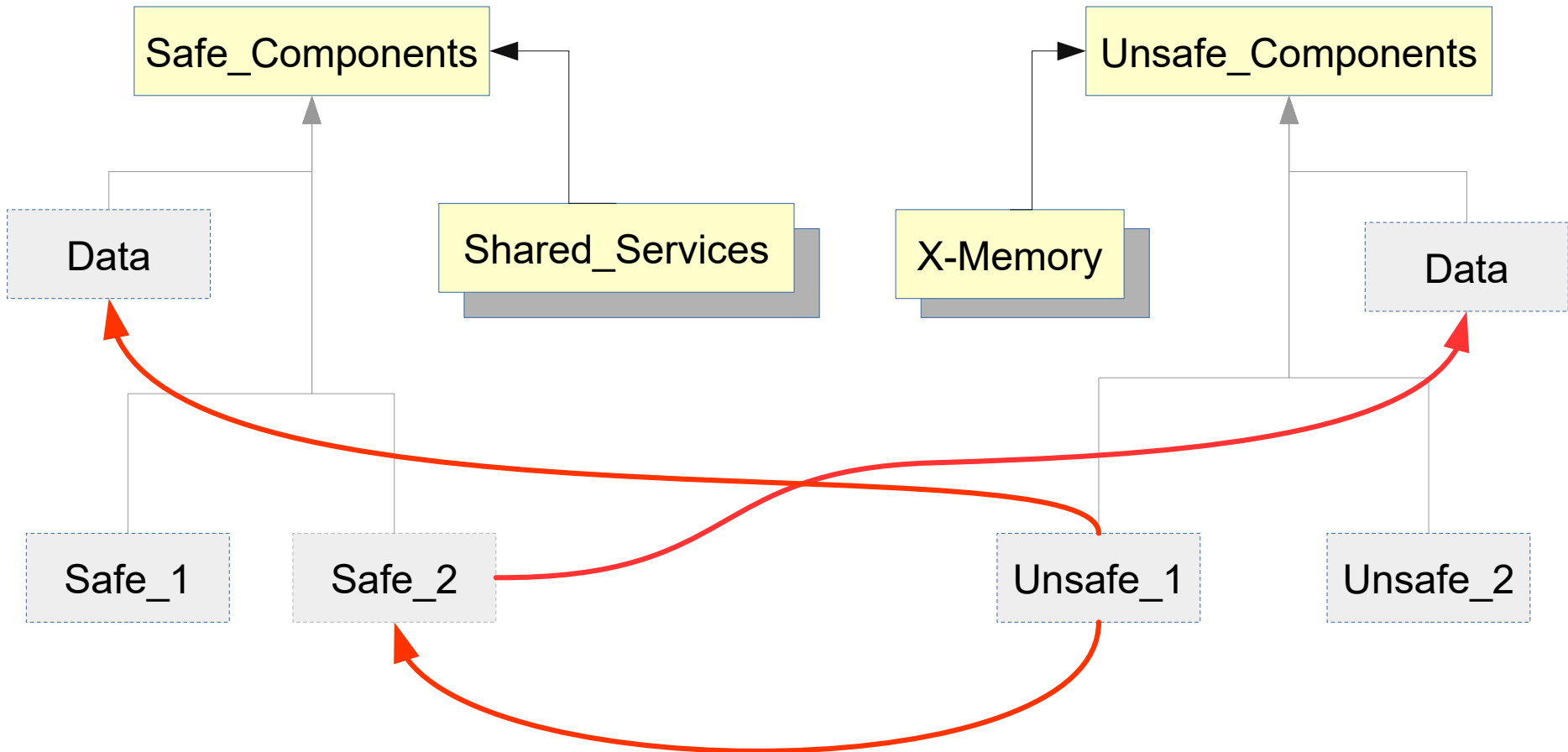


Structure



Public
unit/child


Private child





Other Checks



- Prevent users from cheating with the rules !
 - Requires static analysis
- Use of AdaControl 
 - Free tool provided by Adalog : www.adacontrol.fr
- Ensures :
 - No unchecked programming
 - Can't be hidden in Ada
 - No removal of language checks, including in SIL0 components
 - No visible variable in package specifications



Achievements



- Criticality of a component is immediately identifiable from its full name
 - The name defines applicable rules
 - Cross-criticality accessors are easily identified
- The most important rules of segregation are enforced by proper usage of language features
 - **Violations don't compile!**
- Simple static analysis demonstrates that there is no cheating with the rules

Name another language that can achieve that...