AFFIRM

Compilation Strategy
March 11, 2015

Compilation
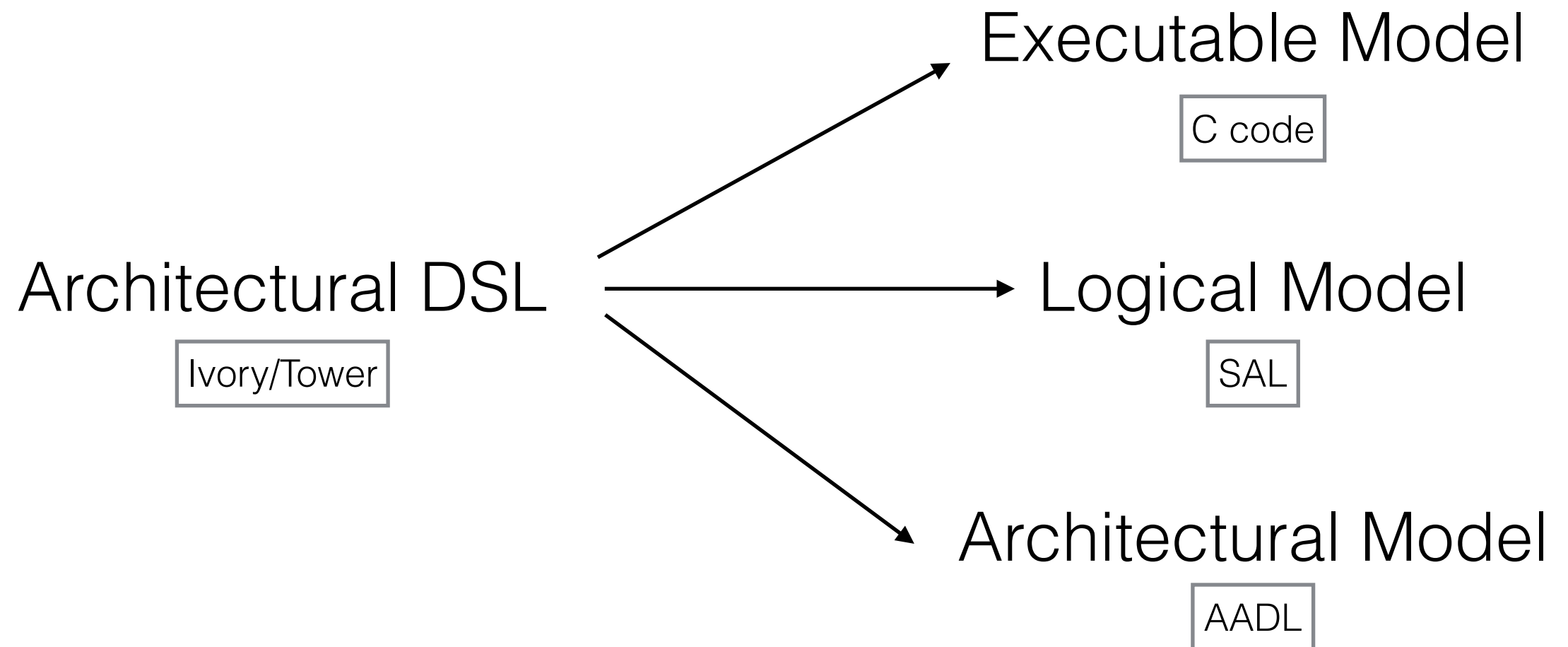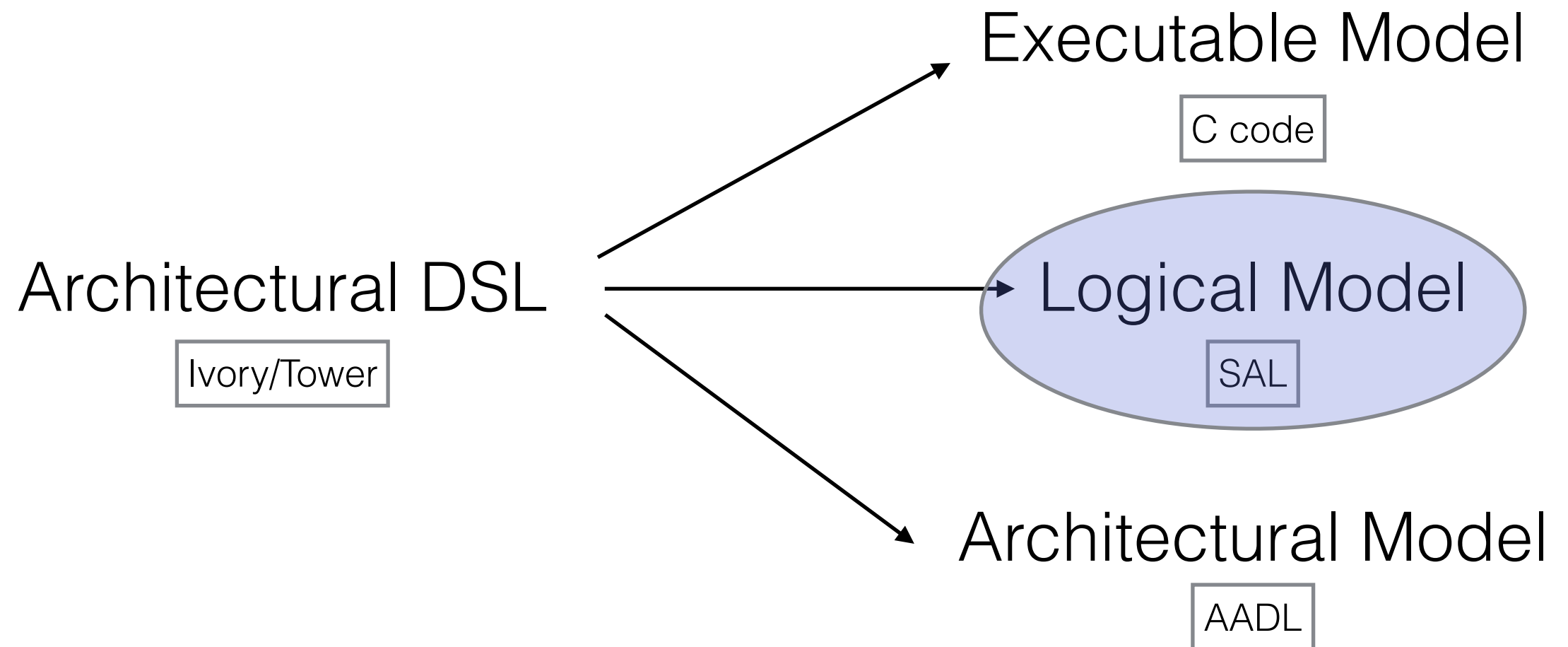
Architectural DSL → Executable Model

Ivory/Tower

C code

Architectural DSL → Logical Model

SAL

Architectural DSL → Architectural Model

AADL

Compilation

Executable Model
C code

Architectural DSL
Ivory/Tower

Logical Model
SAL

Architectural Model
AADL

Components

Tower AST:

- Monitors (handlers)   ⟶   SAL Modules (transition rules)

- Channels

- Signals   }   ⟶   Calendar Automata

- Periods (clocks)

Specification:

- Safety   }   ⟶   Synchronous Observer(s)

- Liveness*

Components

Tower AST:

- Monitors (handlers) ⟶ SAL Modules (transition rules)

- Channels
- Signals   } ⟶ Calendar Automata
- Periods (clocks)

Specification:

- Safety
- Liveness*   } ⟶ Synchronous Observer(s)
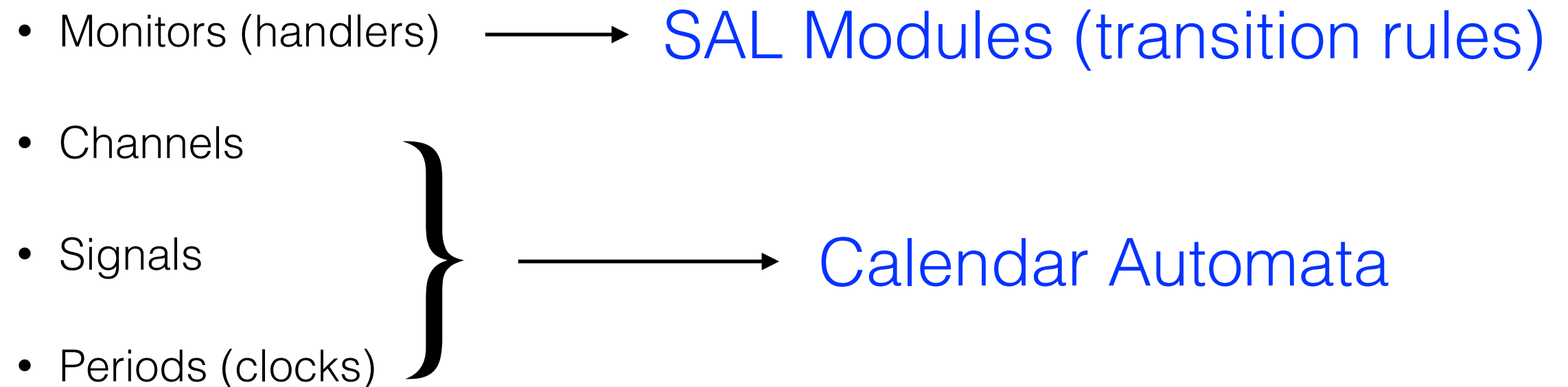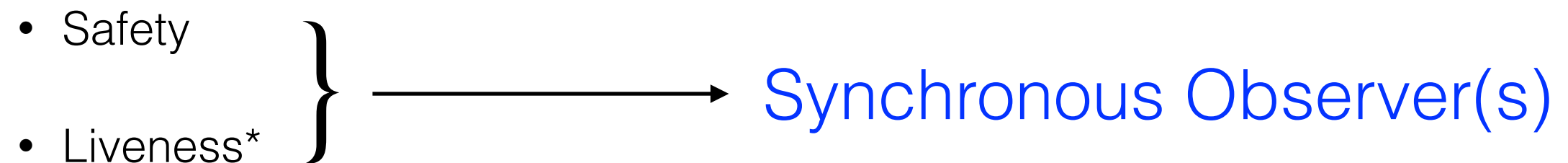
Calendar Automata

- Calendar Automata model time as a **real** variable and always advance to the next time with a scheduled event

- with <u>static calendars</u> we can model periodic clocks and pre-determined signals

- with <u>dynamic calendars</u> we can model message passing over channels

Synchronous Observers

John Rushby's "synchronous observer" is a way of checking a temporal property by simply extending the state machine

- On the ADSL side we can express observers as new <u>Monitor</u>s and give them privileged access to local states of other Monitors

- Using observers we avoid having to construct and interpret a temporal logic in our ADSL

- Observers are <u>composable</u>, and simple ones can be automatically generated