Formal Modeling and Analysis of Stream Processing Systems

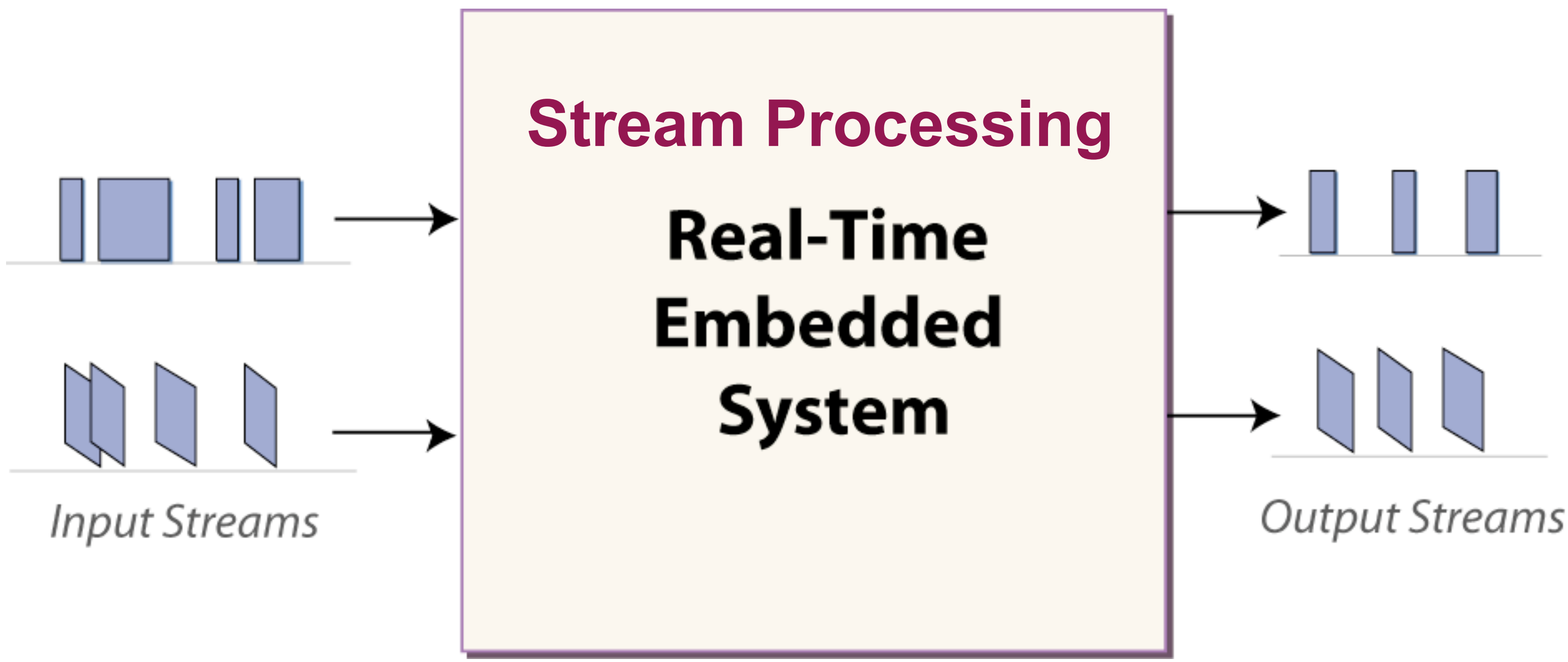
Linh T.X. Phan

March 2009

Penn University of Pennsylvania
Computer and Information Science
University of Pennsylvania

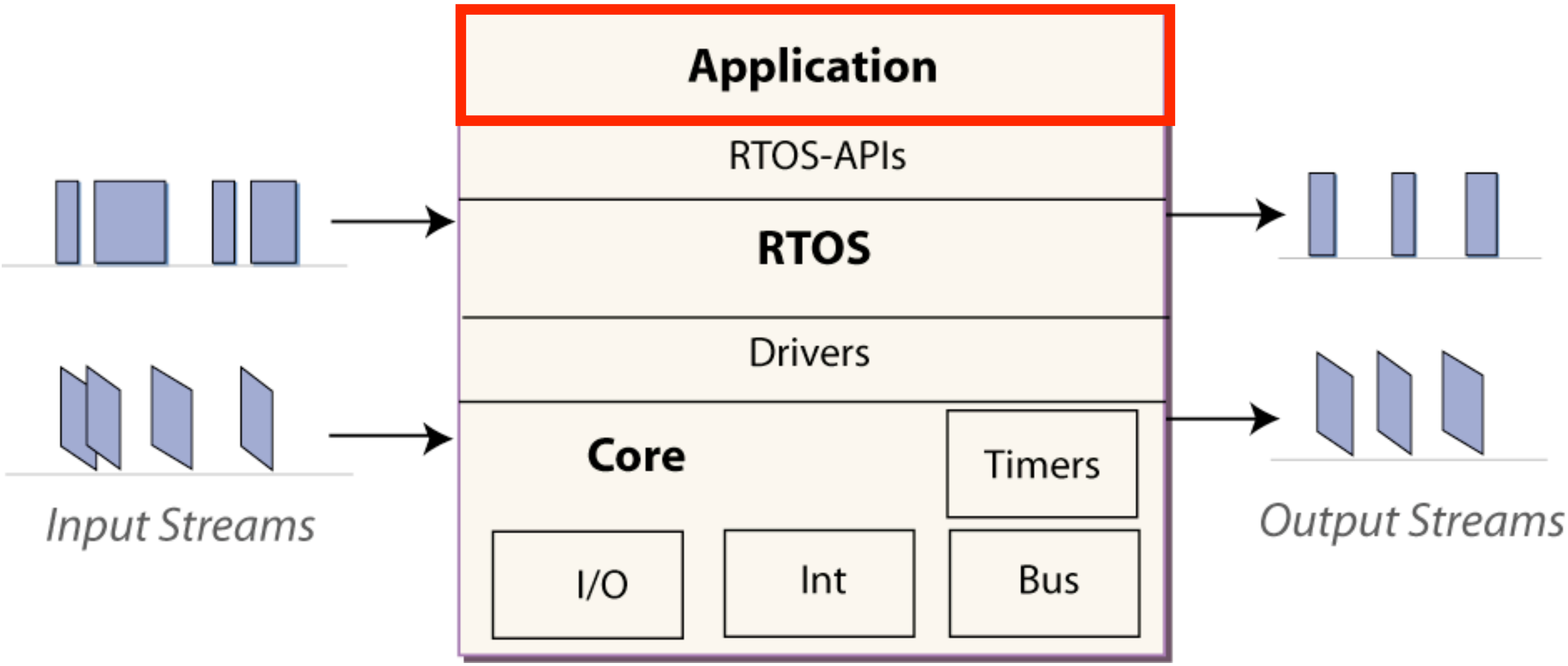

Highly optimized systems
3 Head
ATA66/100
Cable
IDE Interface
5.25" Secure Bay
Secure PCI Adapter
Secure Mobile Rack
MAINBOARD
Hard Drive
Clear Text
Cipher Text
HOSTING
Clients on Internet

Overview

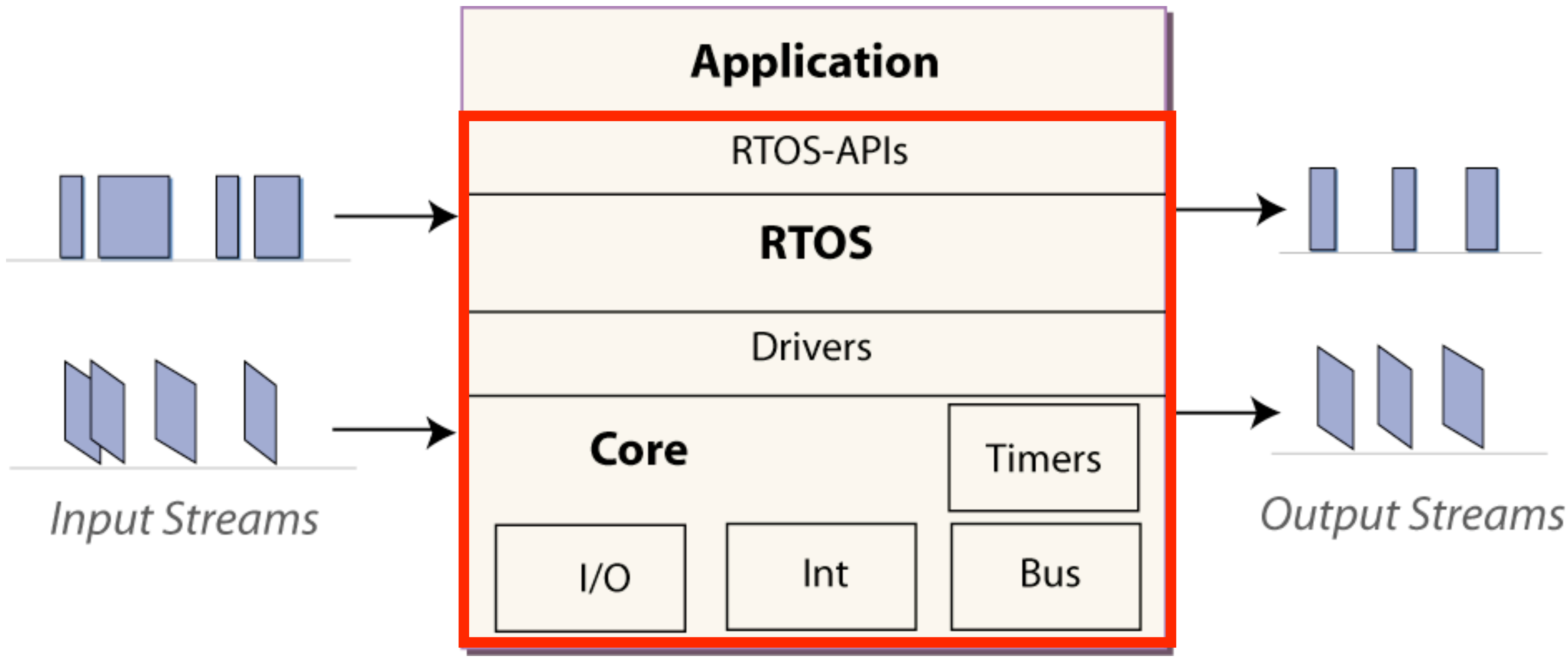


Methods discussed here are applicable to general real-time embedded systems

Inside the box…

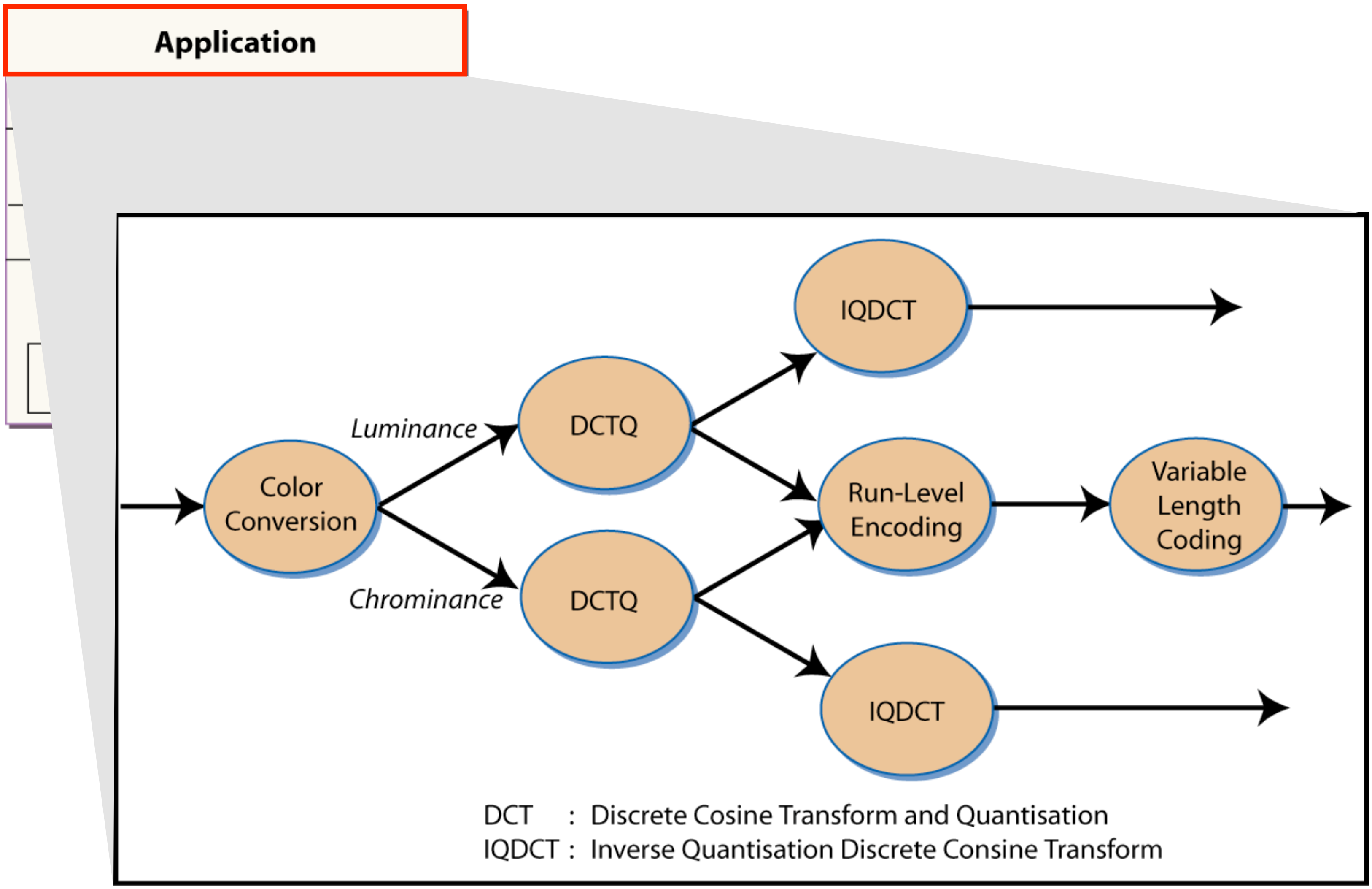


A complete system

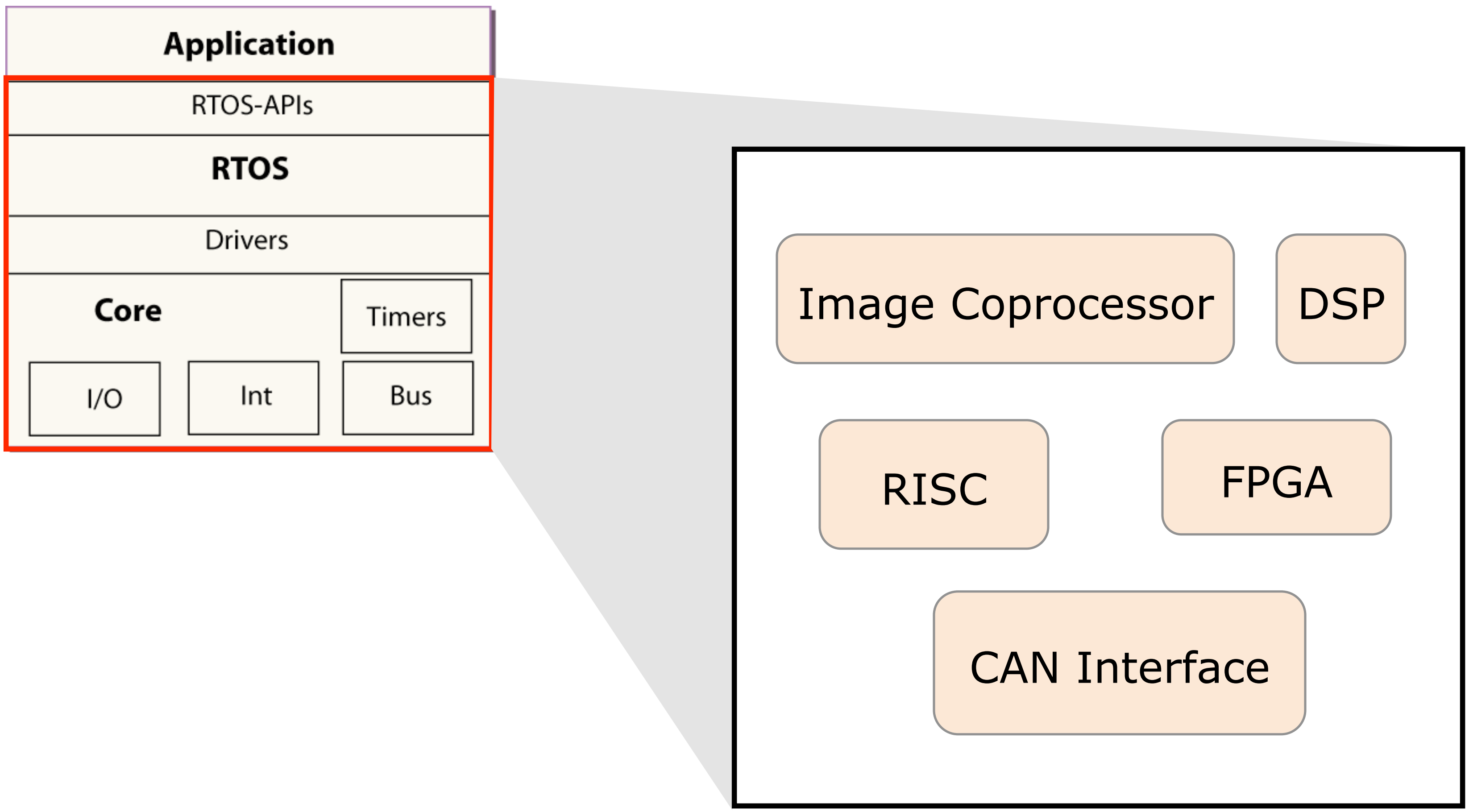


Target Platform

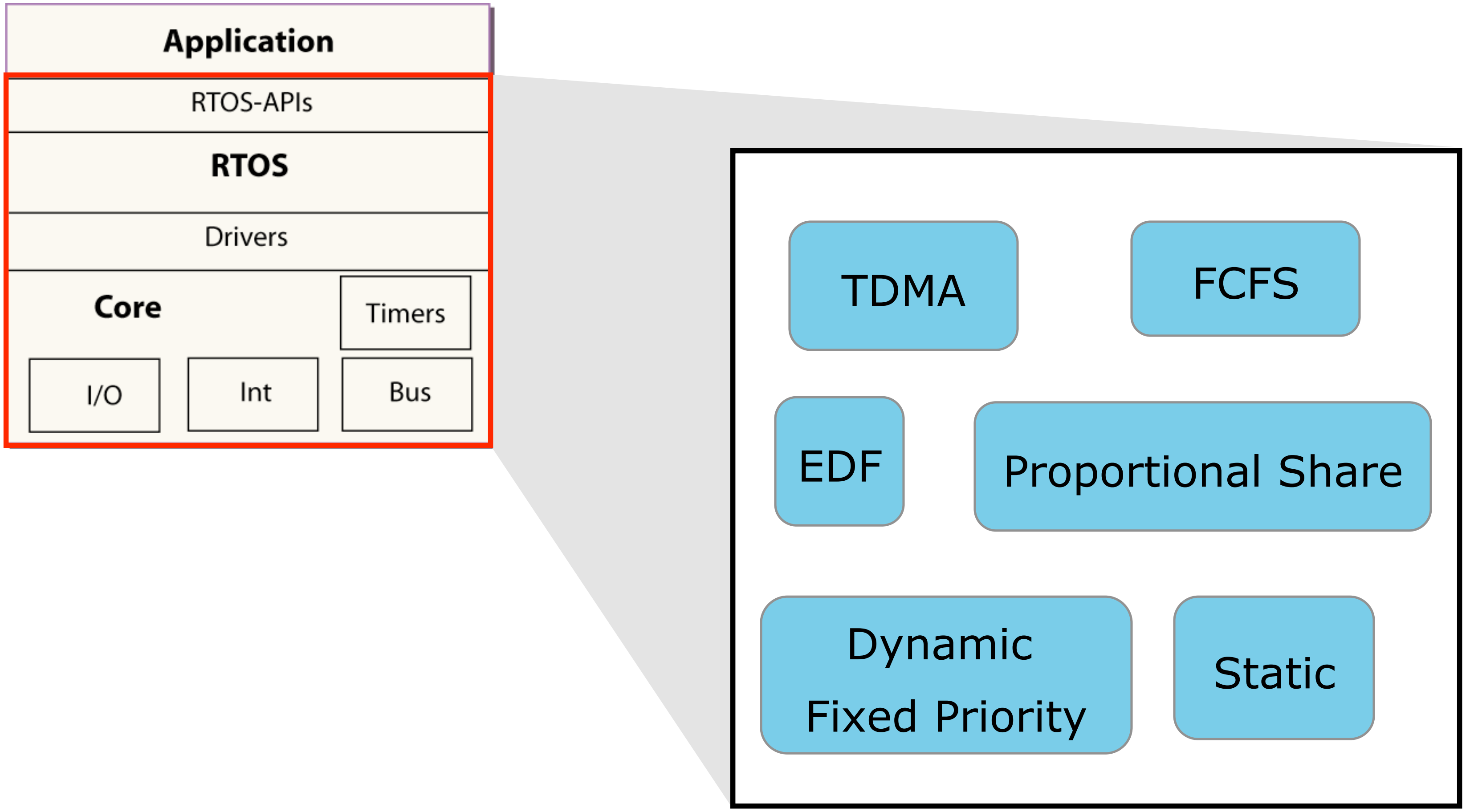
1. Streaming application tasks



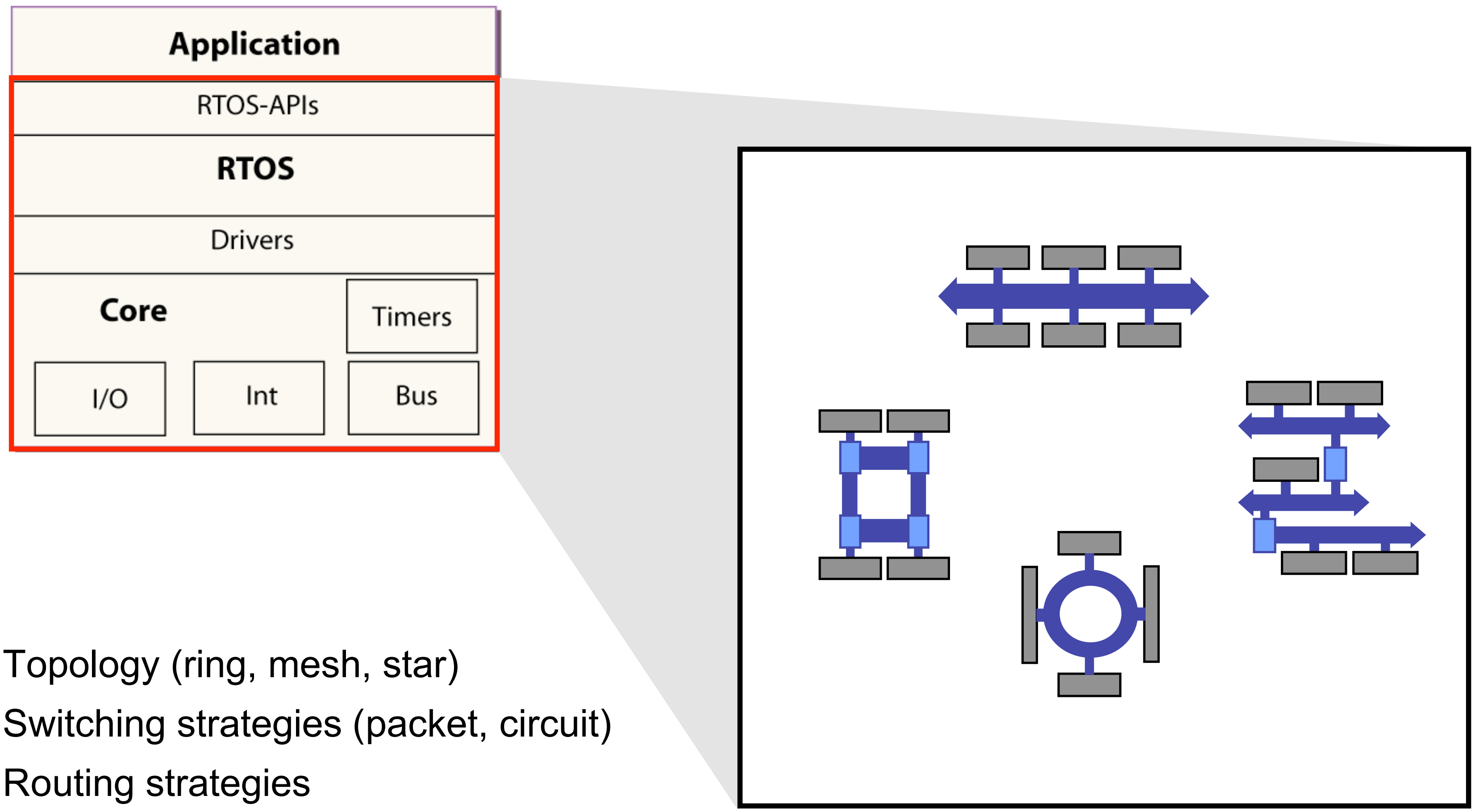
2. Heterogeneous computing and memory resources



3. Heterogeneous RTOS scheduling and synchronization protocols



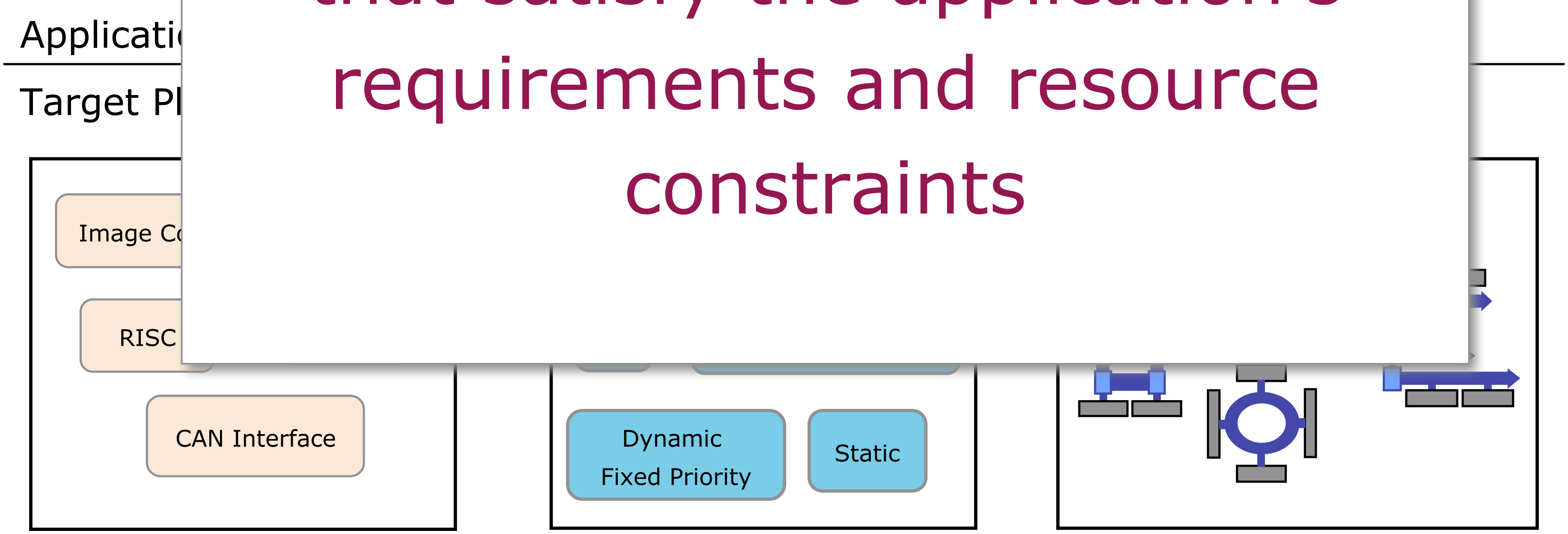
4. Heterogeneous communication resources



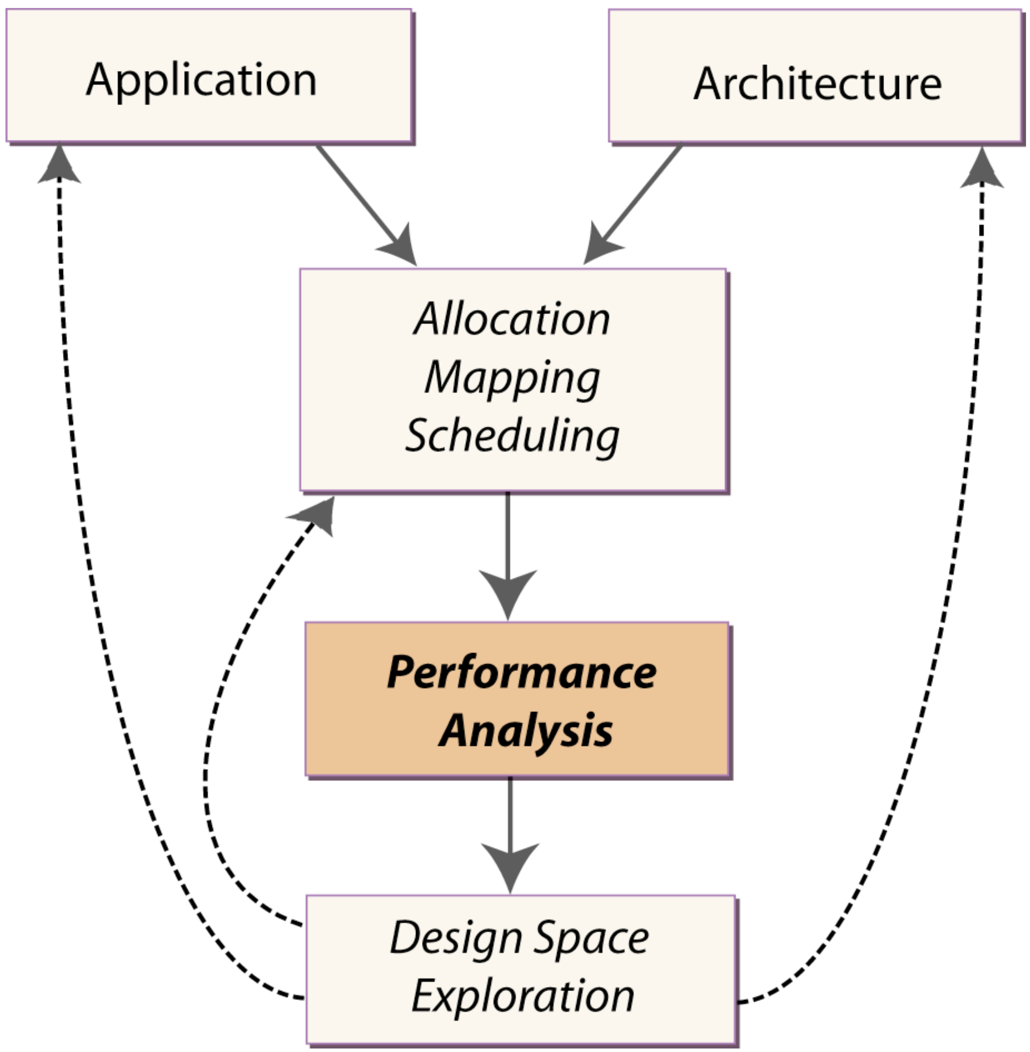
- Topology (ring, mesh, star)
- Switching strategies (packet, circuit)
- Routing strategies
  (static, dynamic, reconfigurable)
- Arbitration policies (dynamic, TDM, CDMA)

The Design Problem

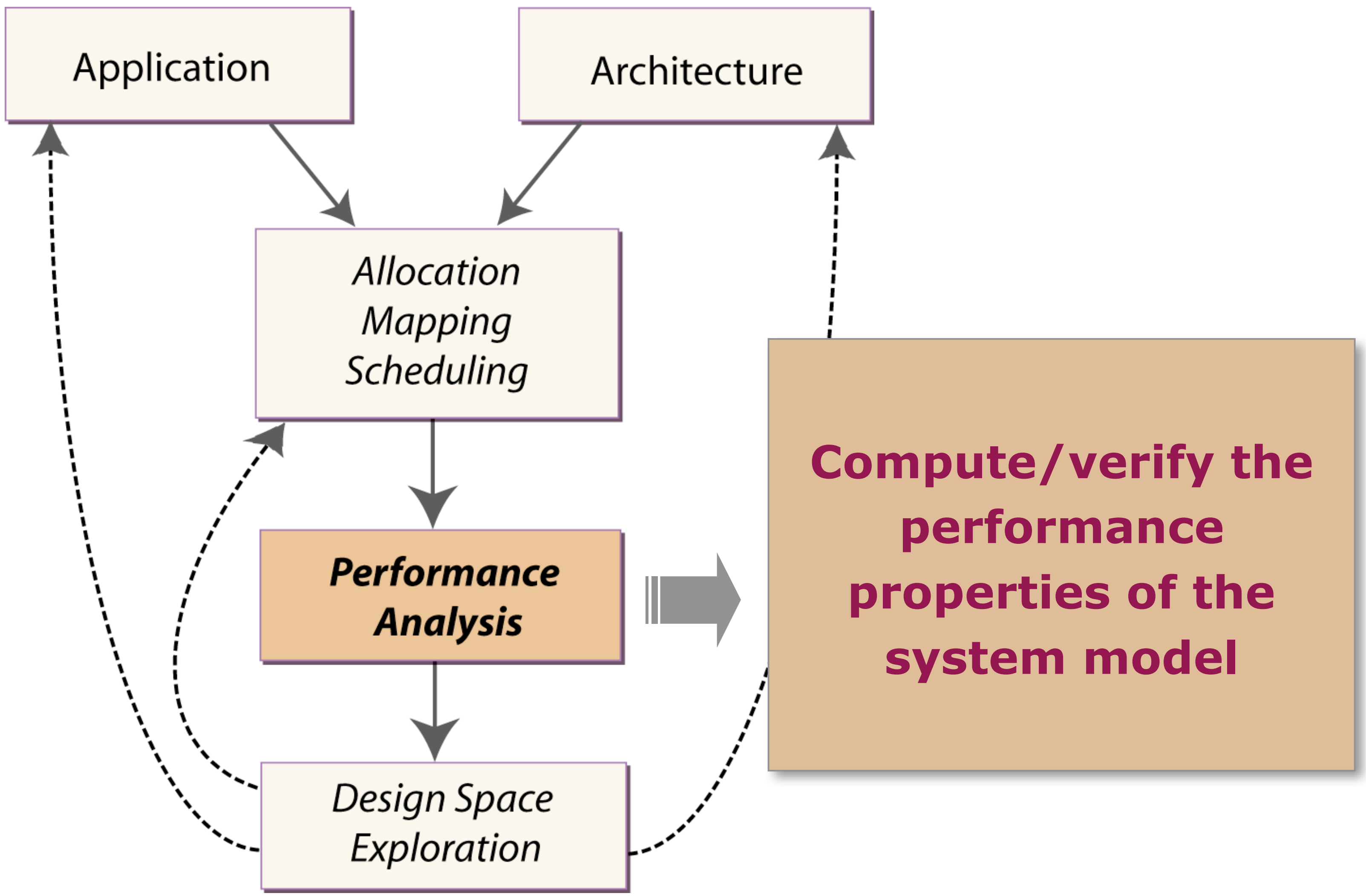
Build a system from subsystems that satisfy the application's requirements and resource constraints



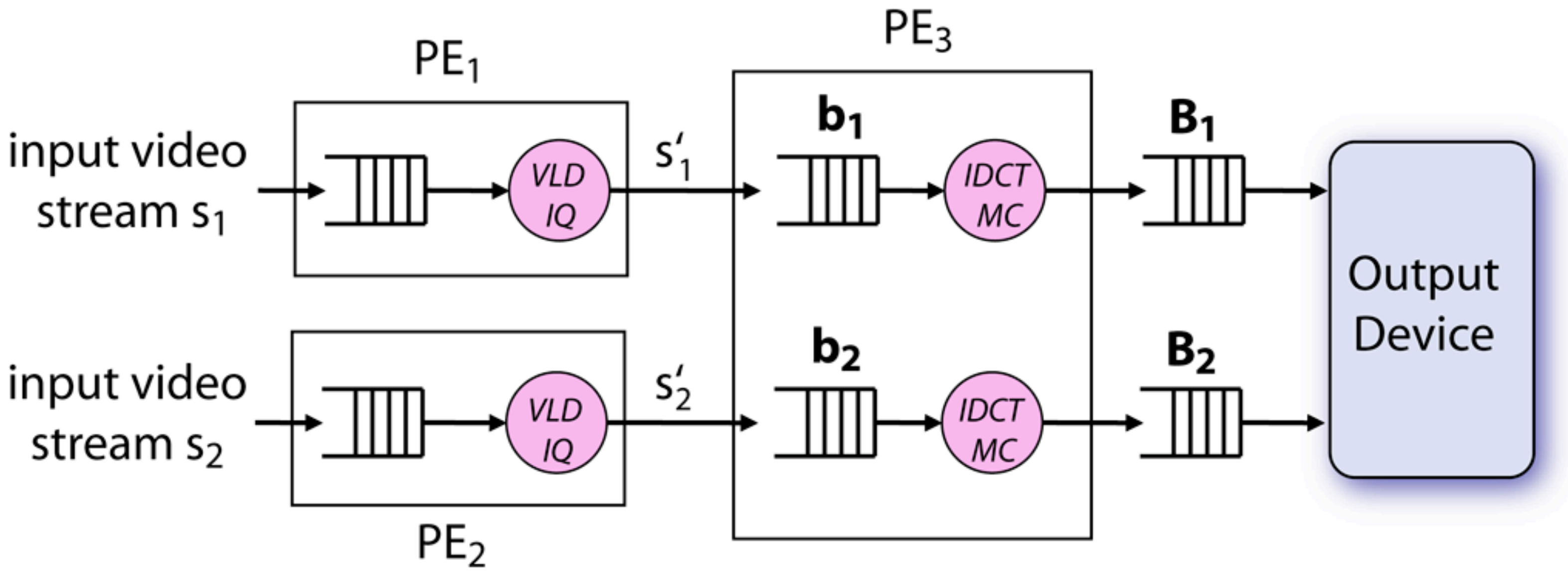
The Design Process



The Performance Analysis Problem



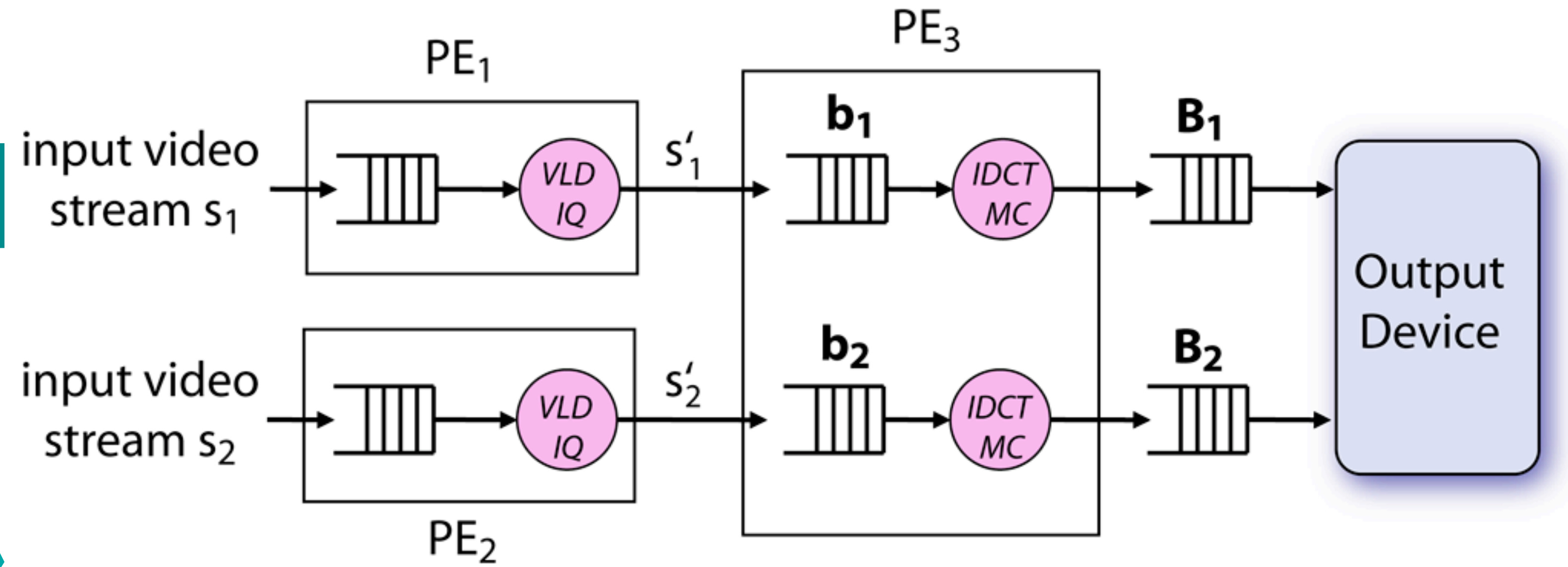
e.g. Architecture of a Picture-in-Picture App.



- Maximum fill-level (backlog) of the buffers?
- Maximum end-to-end delay of the stream?
- Characteristics of the output stream?
- Characteristics of the remaining resource?

. . . . . .

Key Challenges: Complex Event Streams

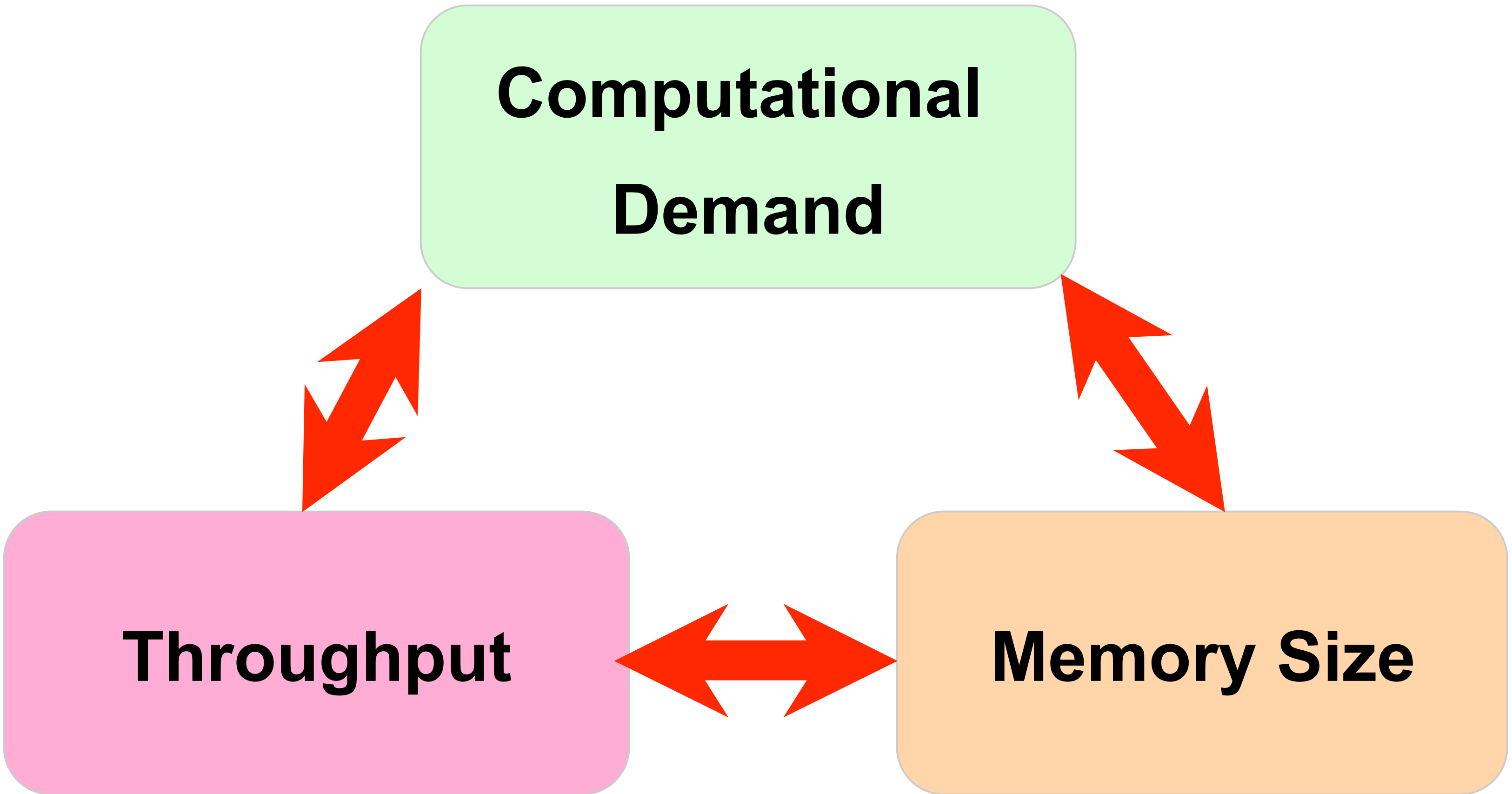


- Infinite sequence of items (events)
- Highly bursty
- Events of multiple types interleaving
- Varied memory and execution demands
- Historically dependent or dynamically controlled

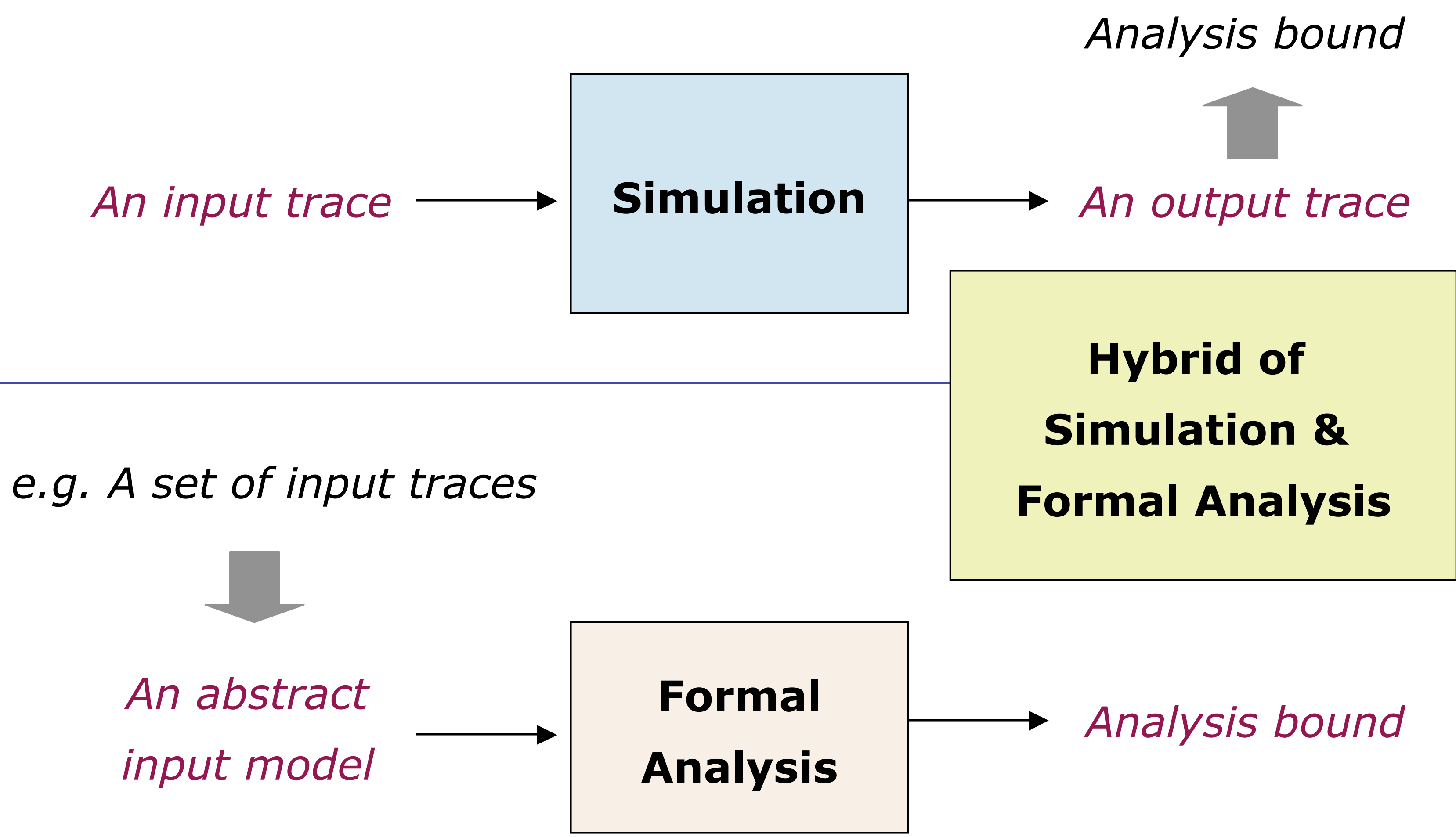
Key Challenges: Complex Tasks & Architectures

- Complex processing semantics
  - fill-level of the buffers
  - execution of an internal automaton
  - synchronization between different streams
- Heterogeneous computing and communication resources
- Various scheduling policies
  - EDF, Fixed-Priority, TDMA, etc.
  - complex state-dependent scheduling schemes

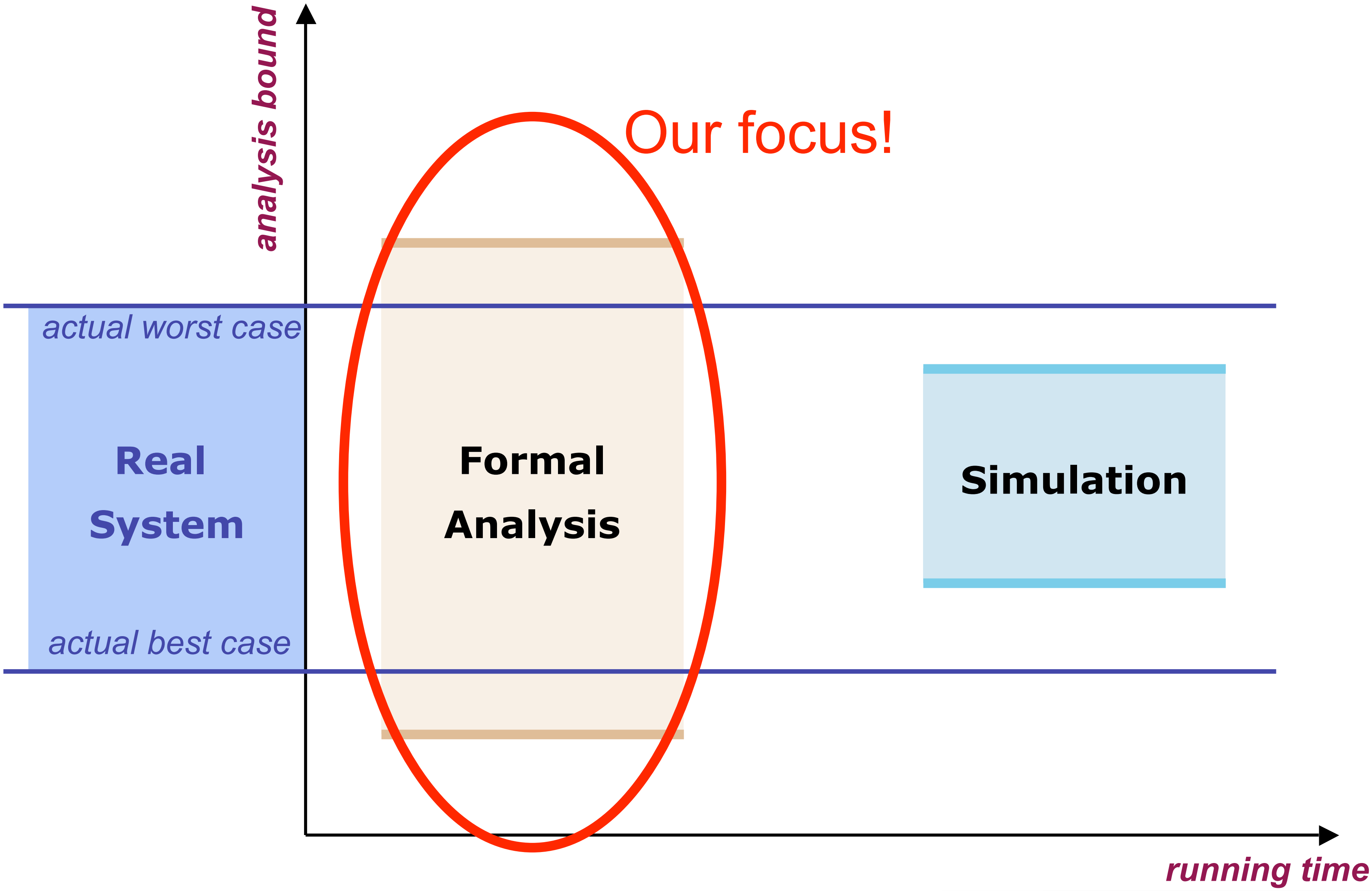
Complex Trade-offs



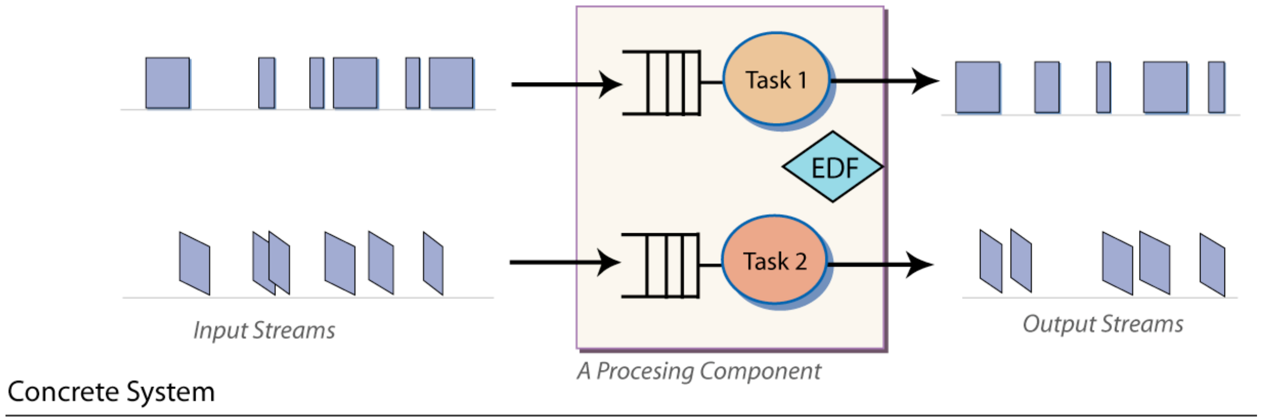
Two Categories of Performance Analysis



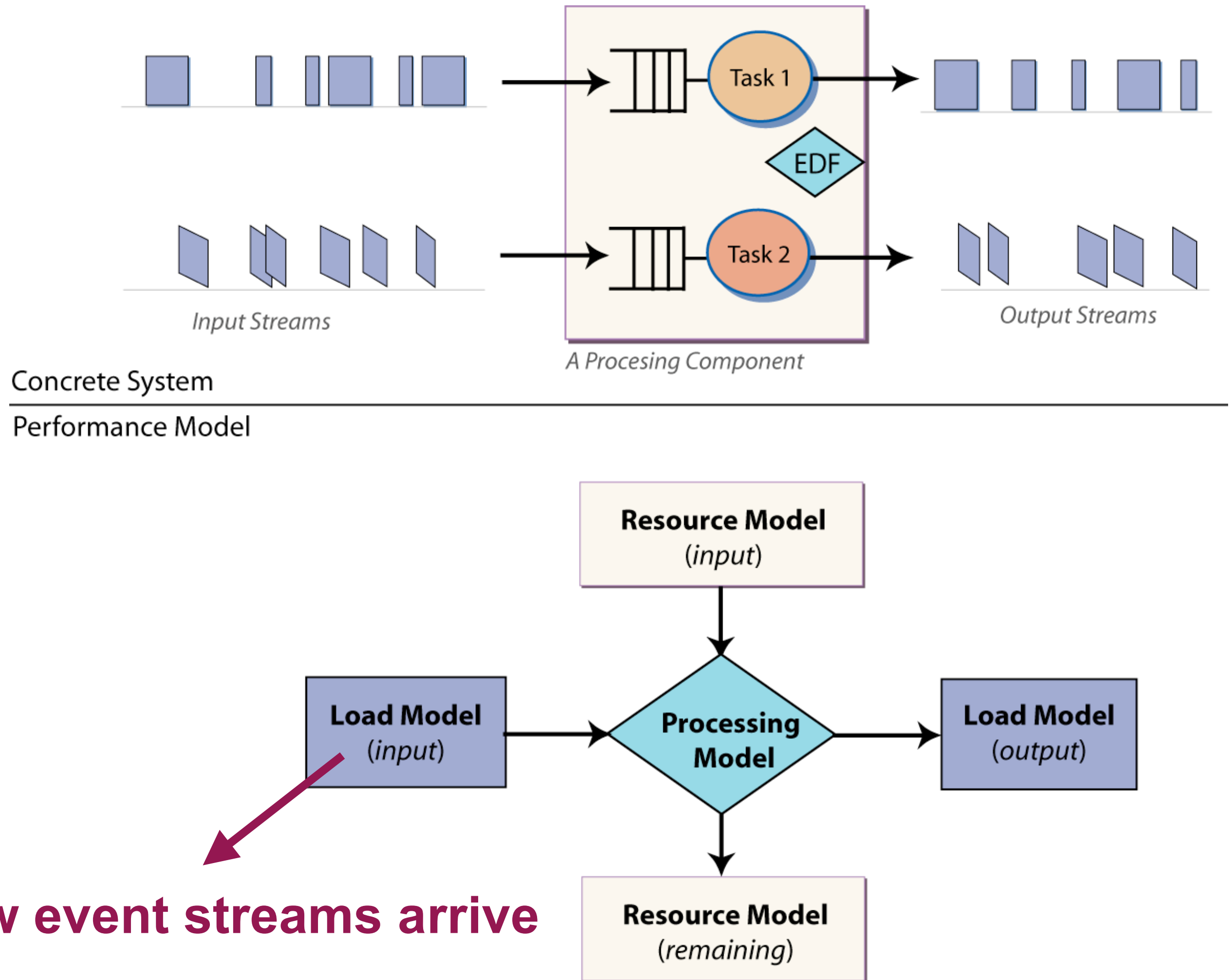
Simulation vs Formal Analysis



Formal Analysis Overview

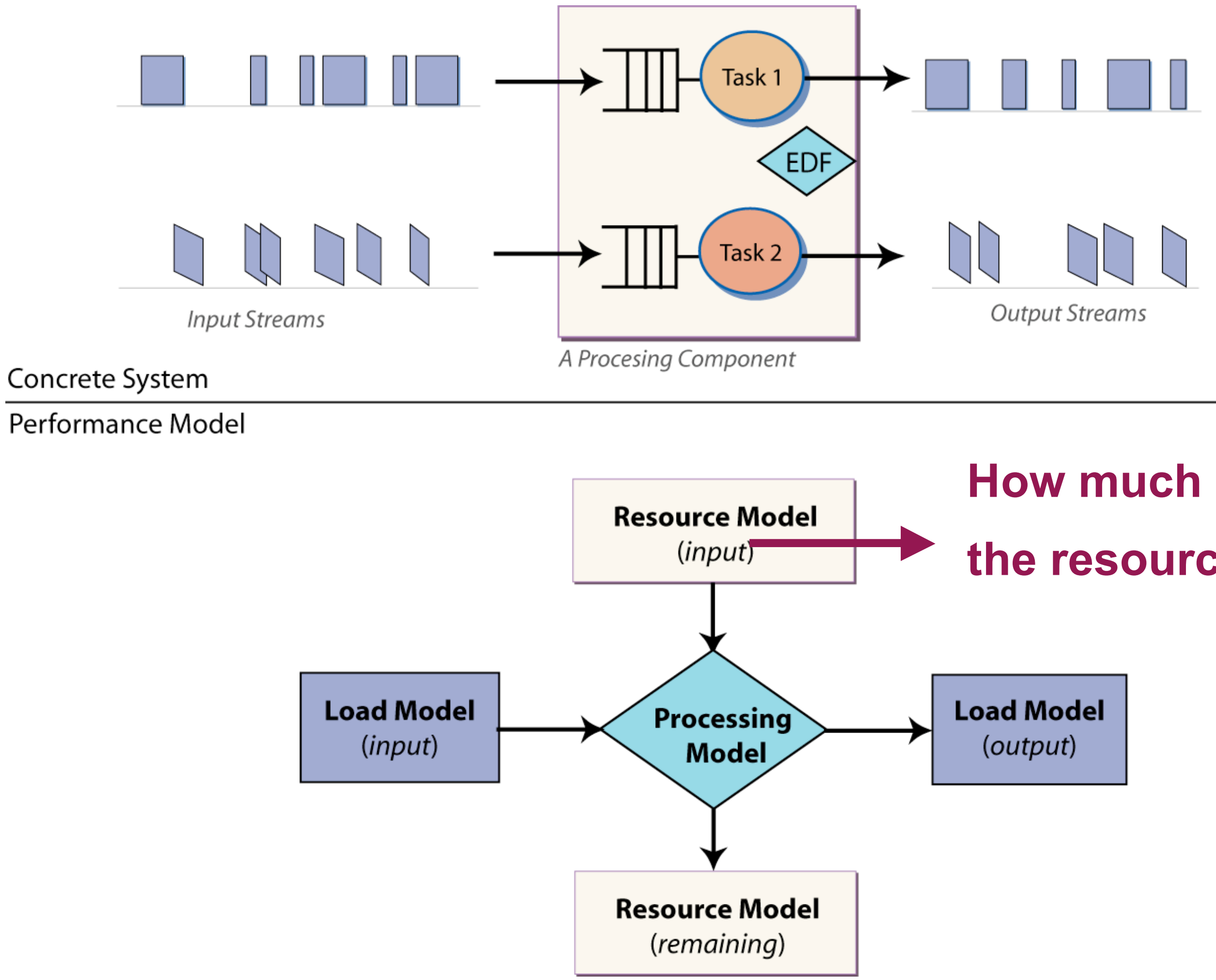


Formal Analysis Overview



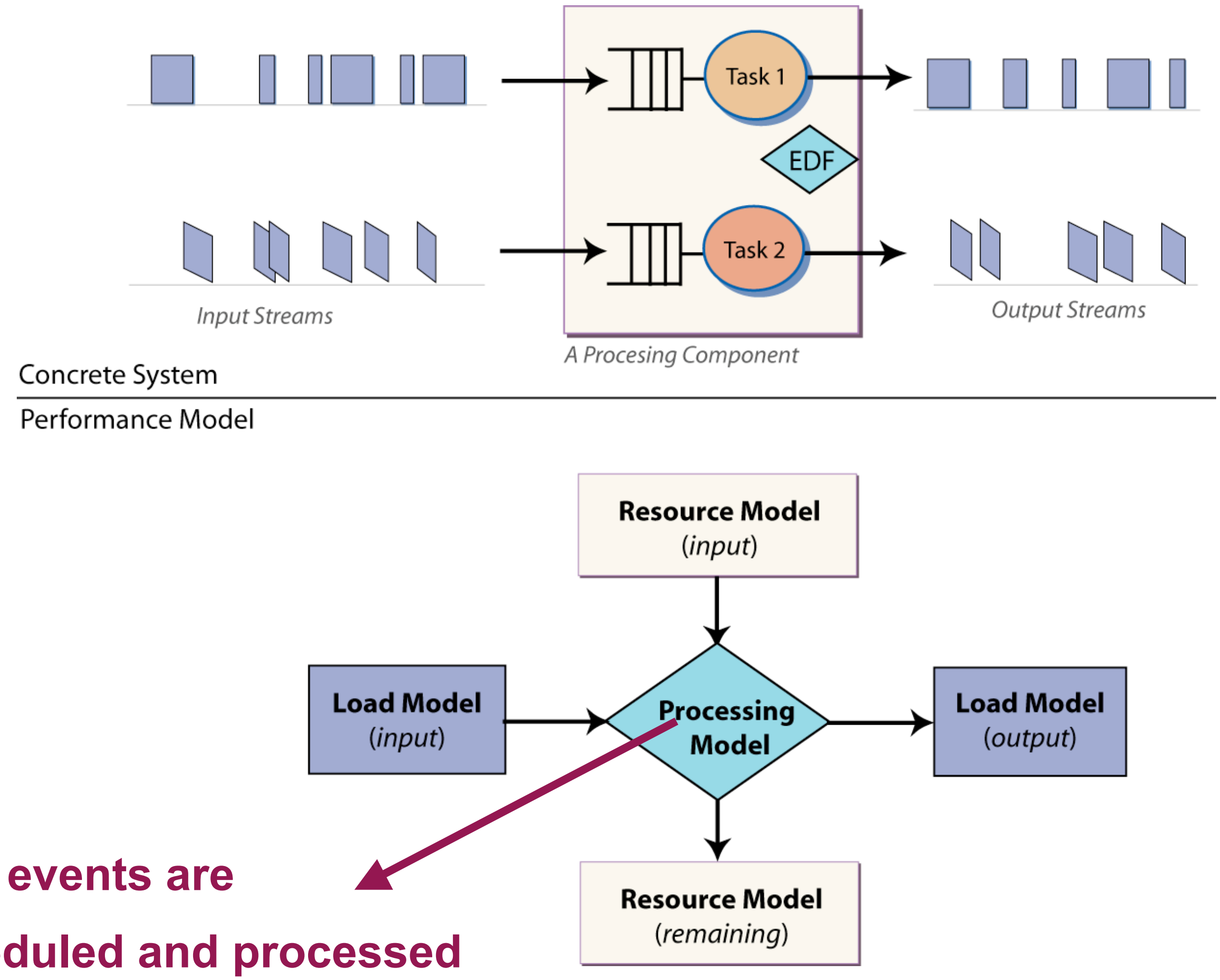
How event streams arrive and its characteristics

Formal Analysis Overview



How much and when the resource are available

Formal Analysis Overview



How events are scheduled and processed

Formal Models and Analysis Methods

1 Standard Event Models (SEM)

– periodic, periodic with jitter/burst and variations

- Simple, easy to analyze
- Unrealistic assumptions
- Too restrictive
- Overly pessimistic results

2 Real-Time Calculus (RTC)

- streams & resources: count-based abstraction
- analysis: (min,+) algebra

Capture burstiness of streams & resource availability

Highly efficient

Cannot model state-dependencies

3 Timed Automata

– model streams at very detailed level
  - capture exact arrival time of each event

Models state-dependencies

Highly accurate

Too detailed → large models

Large systems → inefficient

4 Event Count Automata (ECA)

- syntax: similar to Timed Automata
- semantics: count-based abstraction
  - capture #events in an interval of time

👍 **Models state-dependencies**

👍 **Highly accurate**

👎 **Large systems ⟶ inefficient**

5 Hybrid Models and Methods

- RTC + SEM
- RTC + ECA
- Multi-Mode RTC
- …

👍 **Good accuracy-efficiency trade-off**

The rest of the talk…

Formal Analysis using

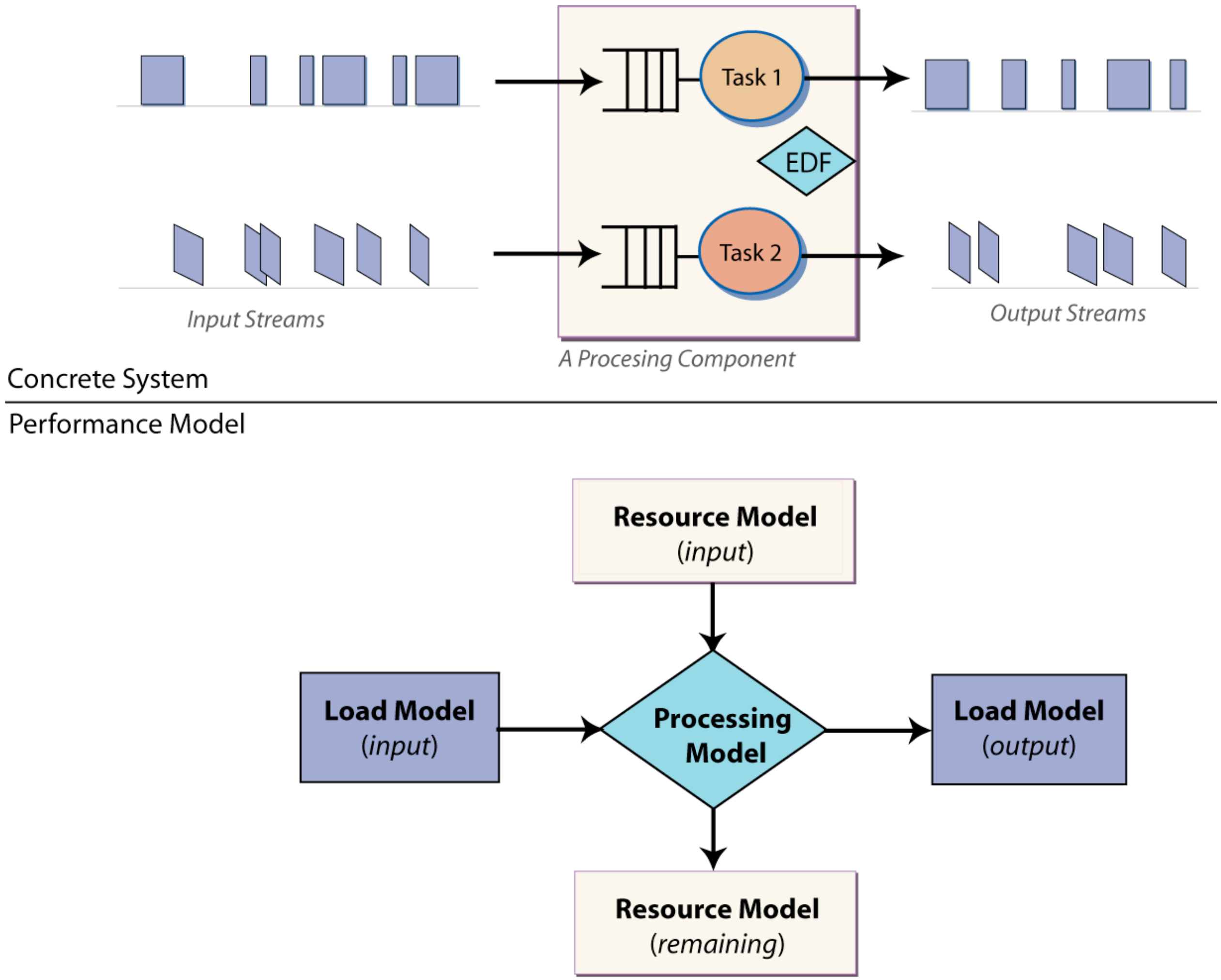
Real-Time Calculus

(RTC)

RTC Background

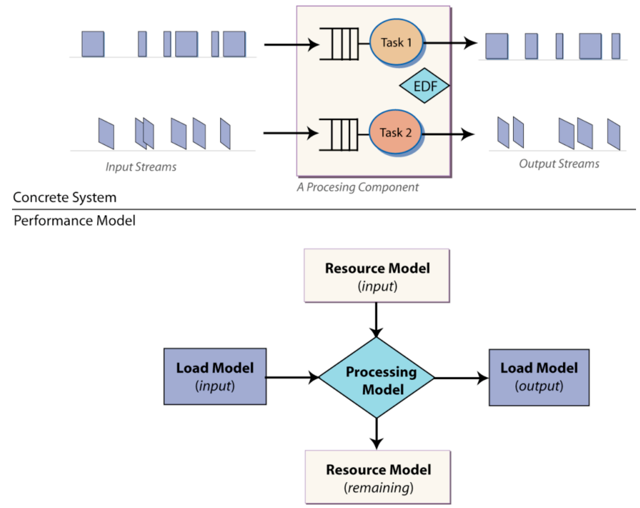
- Originated from Network Calculus in computer networks domain
  - extended for real-time embedded systems
- Worst-case deterministic formal analysis
  - variant of classical queuing theory
- Abstract models: count-based abstraction
- Analysis: min-plus / max-plus algebra

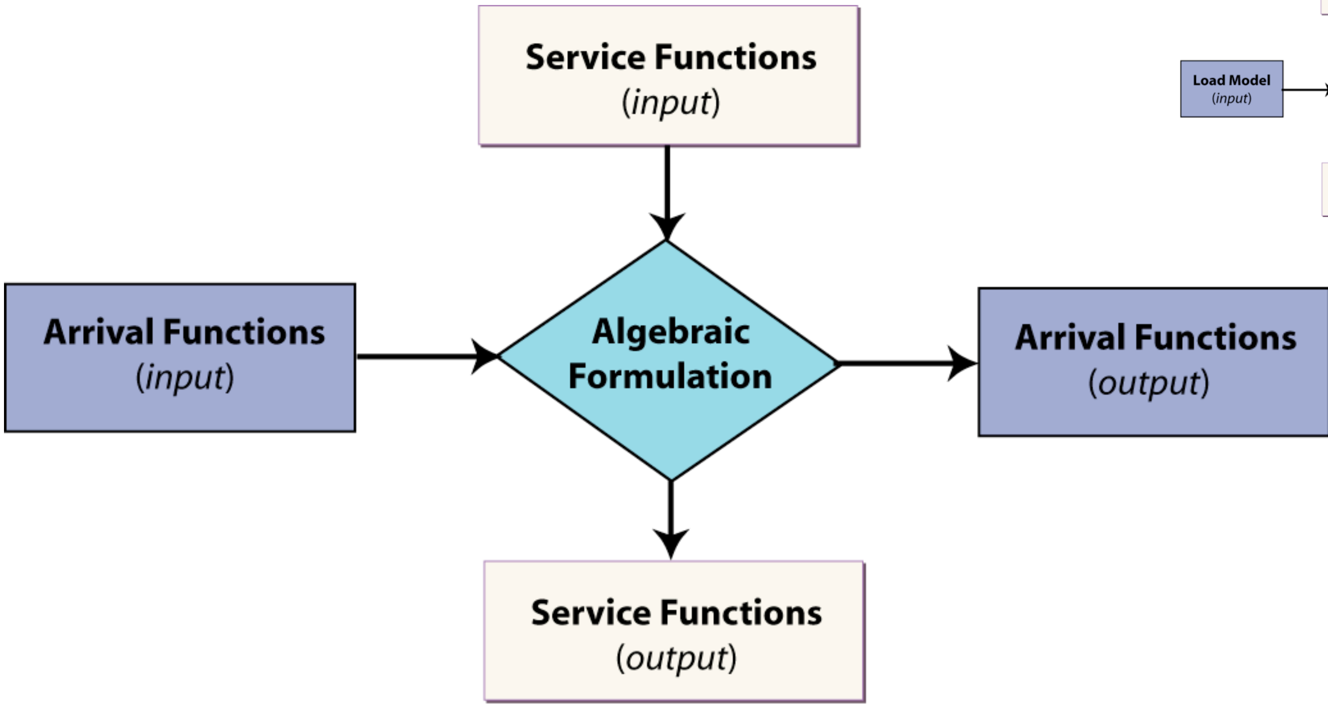
Recall…



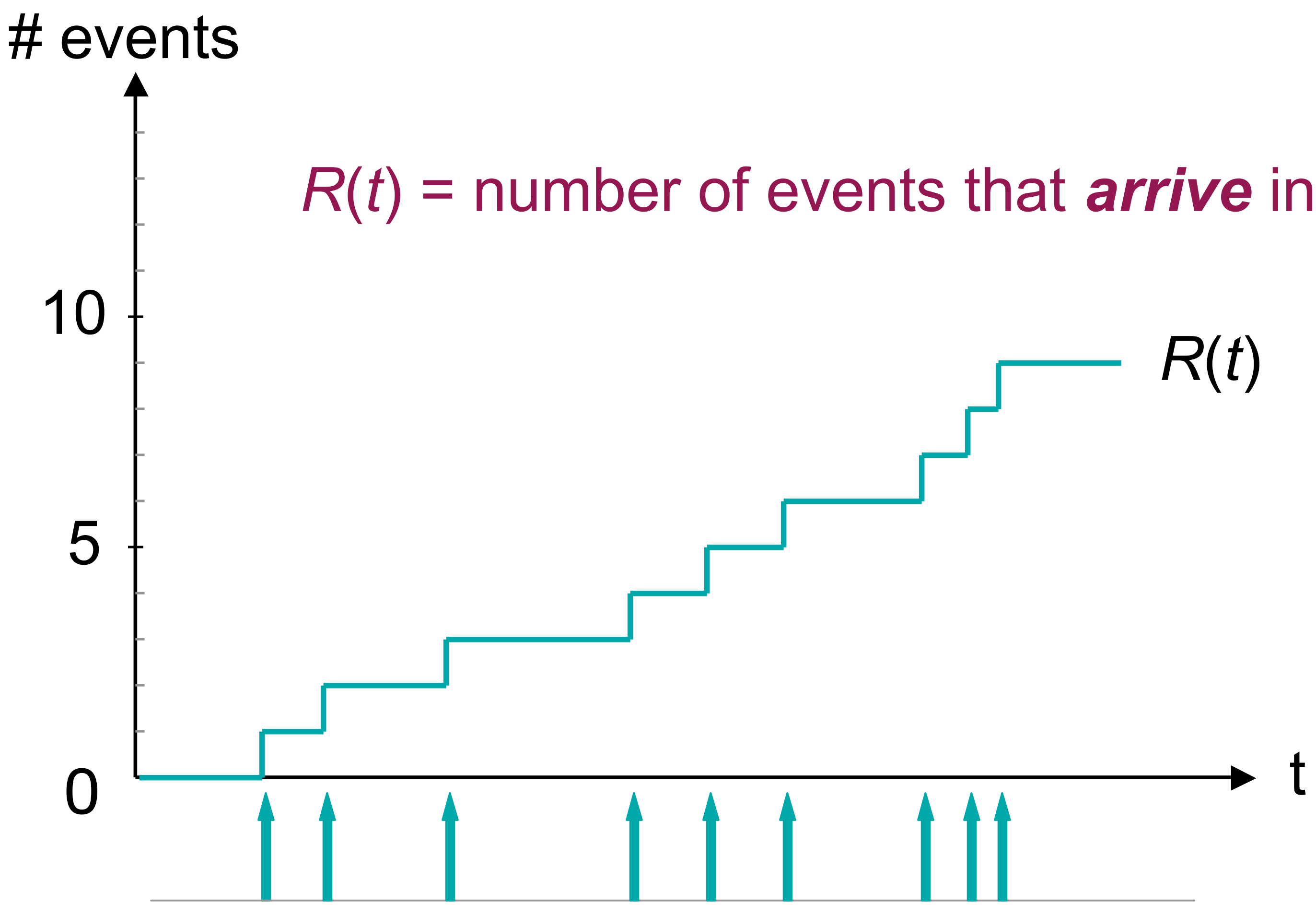
Task 1
EDF
Task 2
Input Streams
Output Streams
A Procesing Component
Concrete System
Performance Model
Resource Model
(input)
Load Model
(input)
Processing
Model
Load Model
(output)
Resource Model
(remaining)

RTC Performance Model





An Arrival Pattern

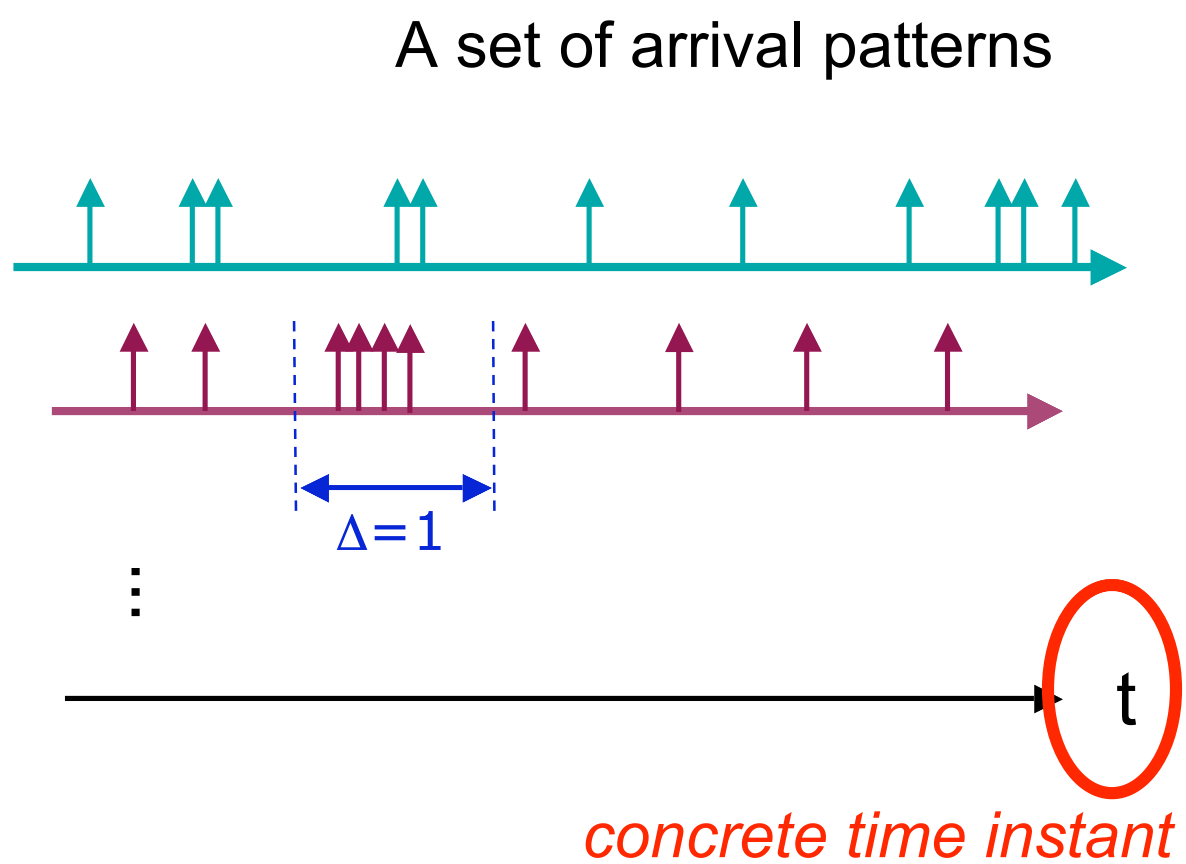


#events that arrive in $[t, t+\Delta)$ is: $R(t+\Delta) - R(t)$

Count-based Abstraction

sliding window size

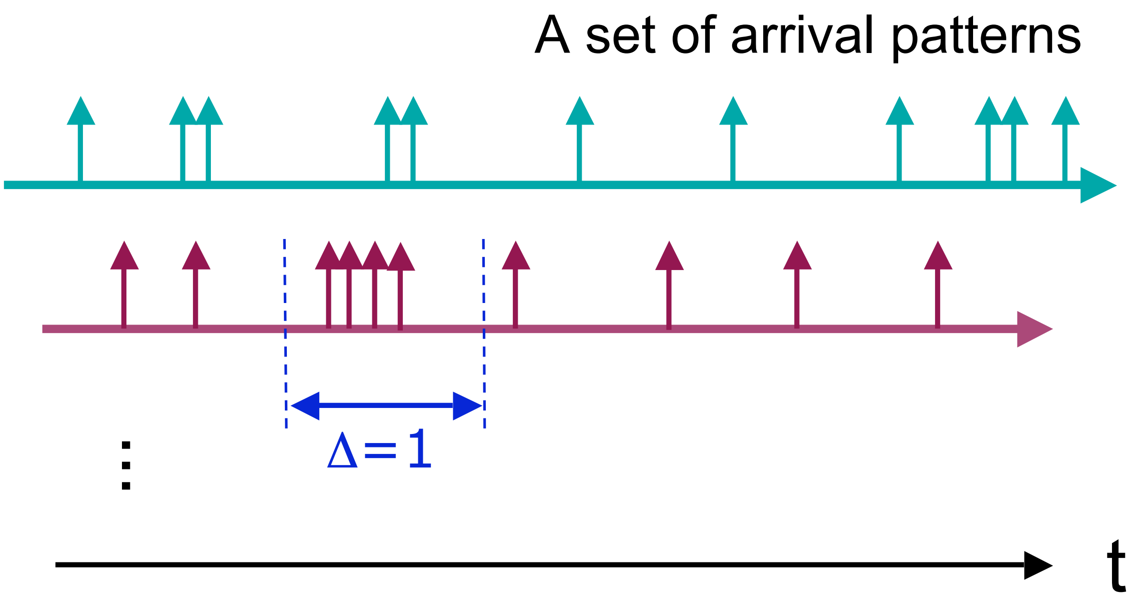
Δ	Lower bound	Upper bound
1	1	4
2	3	6
⋮		



Load Model: Arrival Functions

$\alpha = (\alpha^l, \alpha^u)$

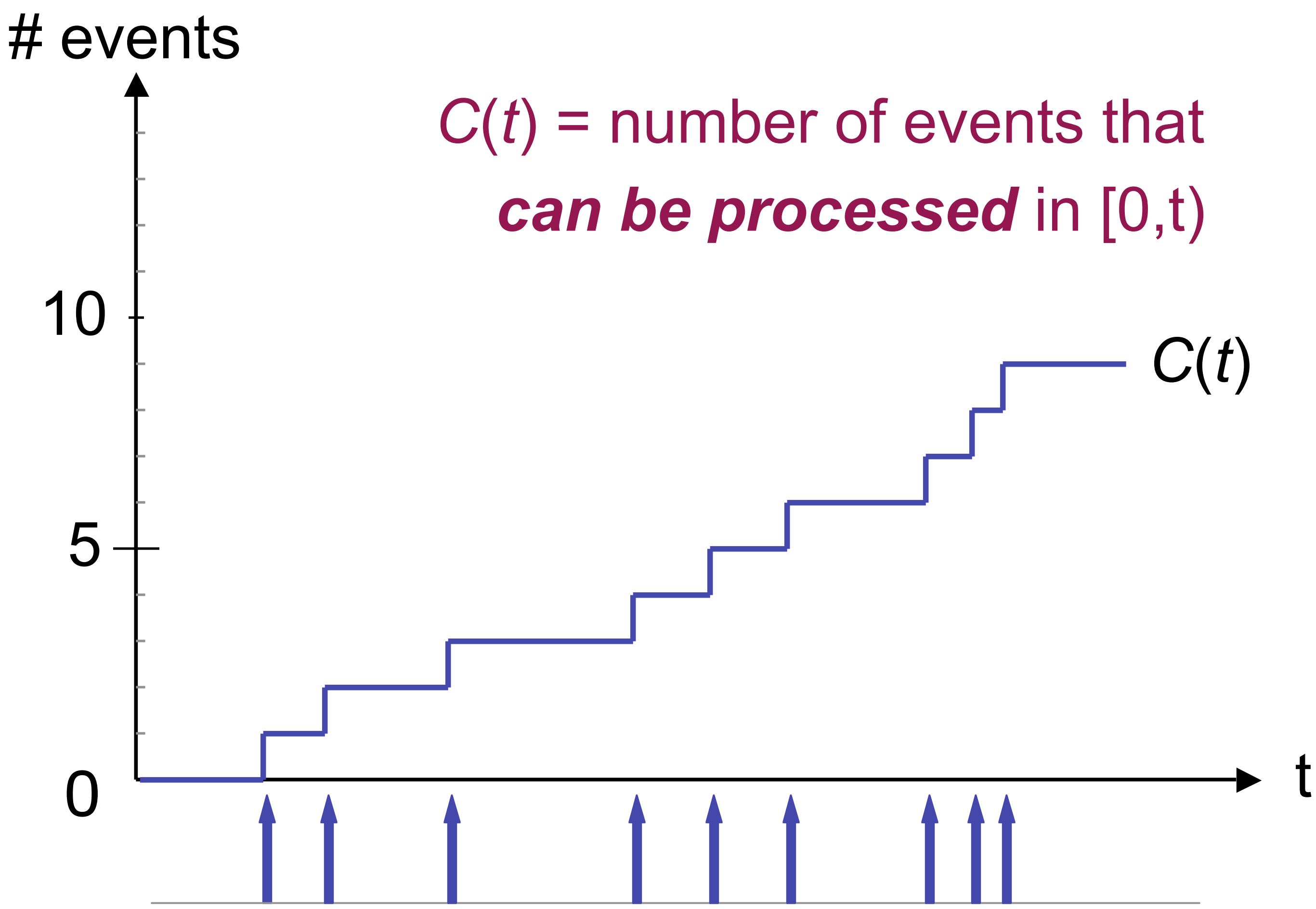
Δ	$\alpha^l(\Delta)$	$\alpha^u(\Delta)$
1	1	4
2	3	6
⋮		



An arrival pattern $R(t)$ satisfies α iff

$$\alpha^l(\Delta) \leq R(t+\Delta) - R(t) \leq \alpha^u(\Delta)$$

A Service Pattern



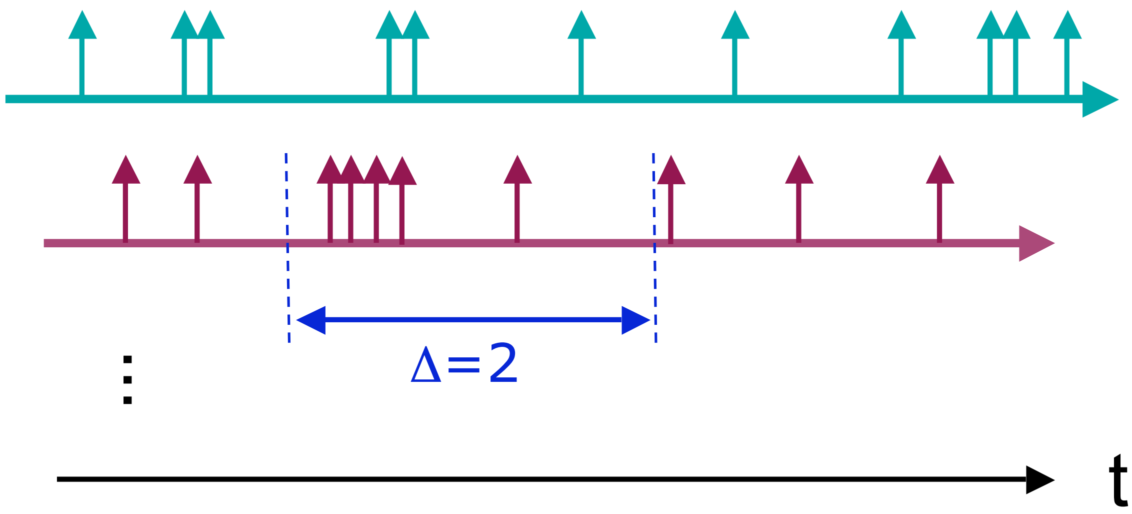
#events that can be processed in [t, t+Δ) is: $C(t+\Delta) - C(t)$

Service Model: Service Functions

$\beta = (\beta^l, \beta^u)$

Δ	$\beta^l(\Delta)$	$\beta^u(\Delta)$
1	0	3
2	3	5
⋮		

A set of service patterns



A service pattern $C(t)$ satisfies β iff

$$\beta^l(\Delta) \leq C(t+\Delta) - C(t) \leq \beta^u(\Delta)$$

Units of Arrival and Service Functions

- $[R(t), \alpha(\Delta)]$ and $[C(t), \beta(\Delta)]$ can also be specified in terms of the number of resource units
  - processor cycles, transmiting bit, etc.

- Should *always* convert to the same unit before performing analysis