

# Distributed Runtime

Cheoljoo Jeong

# Distributed Runtime for The Meadow System

The Meadow system consists of many *devices* which contains *processes* reactive to *events*.

Each device includes a **runtime** which is responsible for routing events between devices and schedule the execution of processes upon arrival of events.

A process can have a *state* and the states of processes are maintained by the runtime<sup>1</sup>.

Runtimes collectively implements “*global lookup tables*.”

---

<sup>1</sup>Or should we assume that the process body is non-time-consuming?

# Workflow Graphs

A user provides *workflow definitions*, which will be compiled into *workflow graphs*.

A **workflow graph** is  $G = (V, E)$ , where  $V$  is a set of reactive processes and  $E$  is a set of directed, predicated edges between processes.

The runtime in the device is responsible for *execution of workflow graphs*, which involves 1) launching of processes and 2) routing of event instances to other runtime.

## Workflow Graphs (Cont.)

A **vertex** of a work process represents a process. A workflow process performs one of the following activities:

- **Computation:** This happens usually execution of the “passive callbacks” by the runtime.
- **Event instance generation:** An event instance is generated, which will be routed by the runtime.

A **directed edge** of a workflow graph is  $(u, v, C)$ , where  $u, v$  are indicient vertices and  $C$  is a condition over the events to which the graph is sensitive to.