

# **Soar2Soar**

**Nate Derbinsky**

University of Michigan

# Motivation

Writing a custom SML application is the current Soar environment development paradigm

Pros	Cons
Flexible	Lots to keep track of and prone to [memory] errors
Many common programming languages	Some tasks are not a good fit for imperative programming

# Introducing Soar2Soar

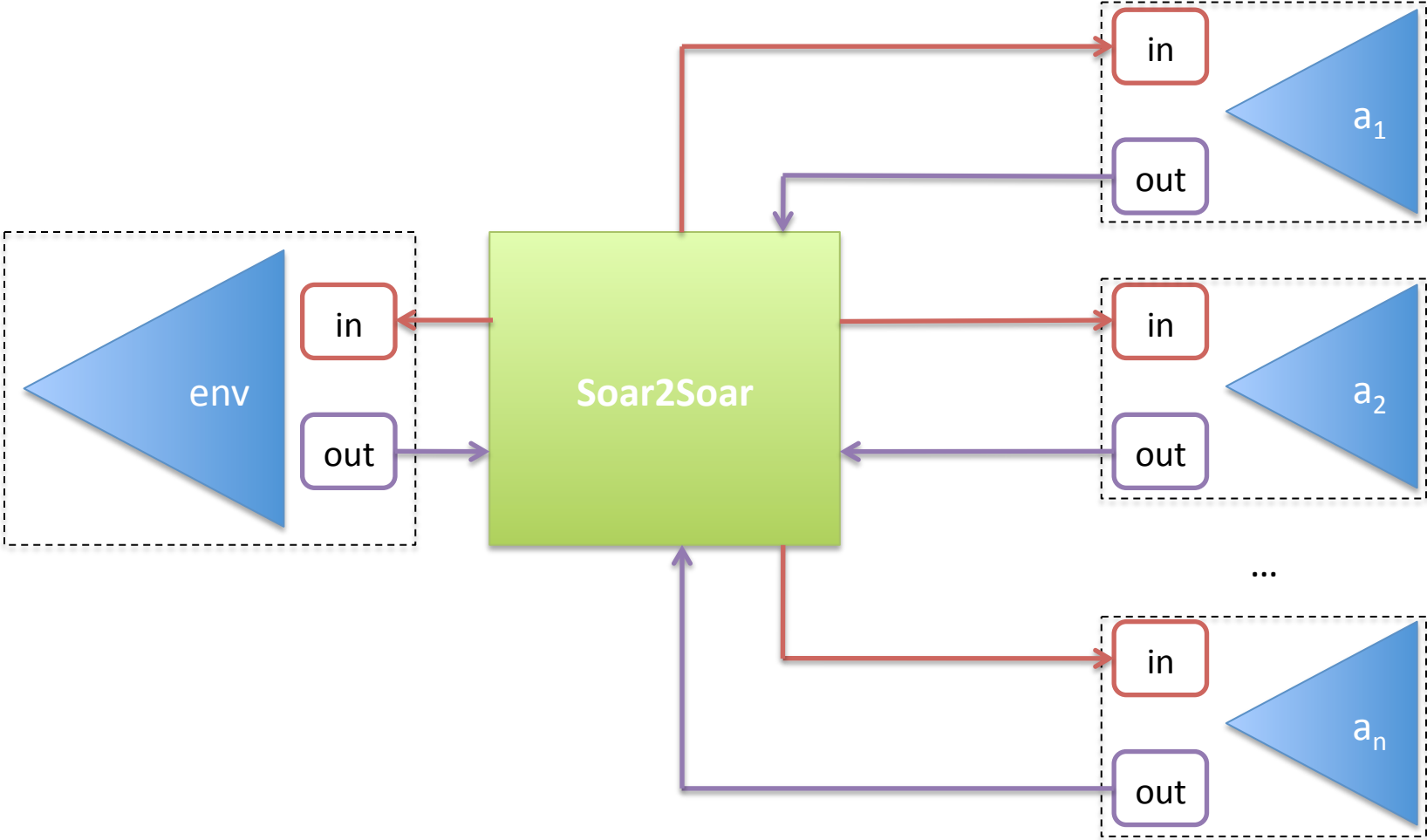
C++ SML middleware that allows a Soar agent to serve as the environment to one or more client Soar agents

- Fast environment prototyping with only Soar
- “Scales” to arbitrary number of client agents
- Released with Soar 9.3.0

# Soar2Soar

## Environment

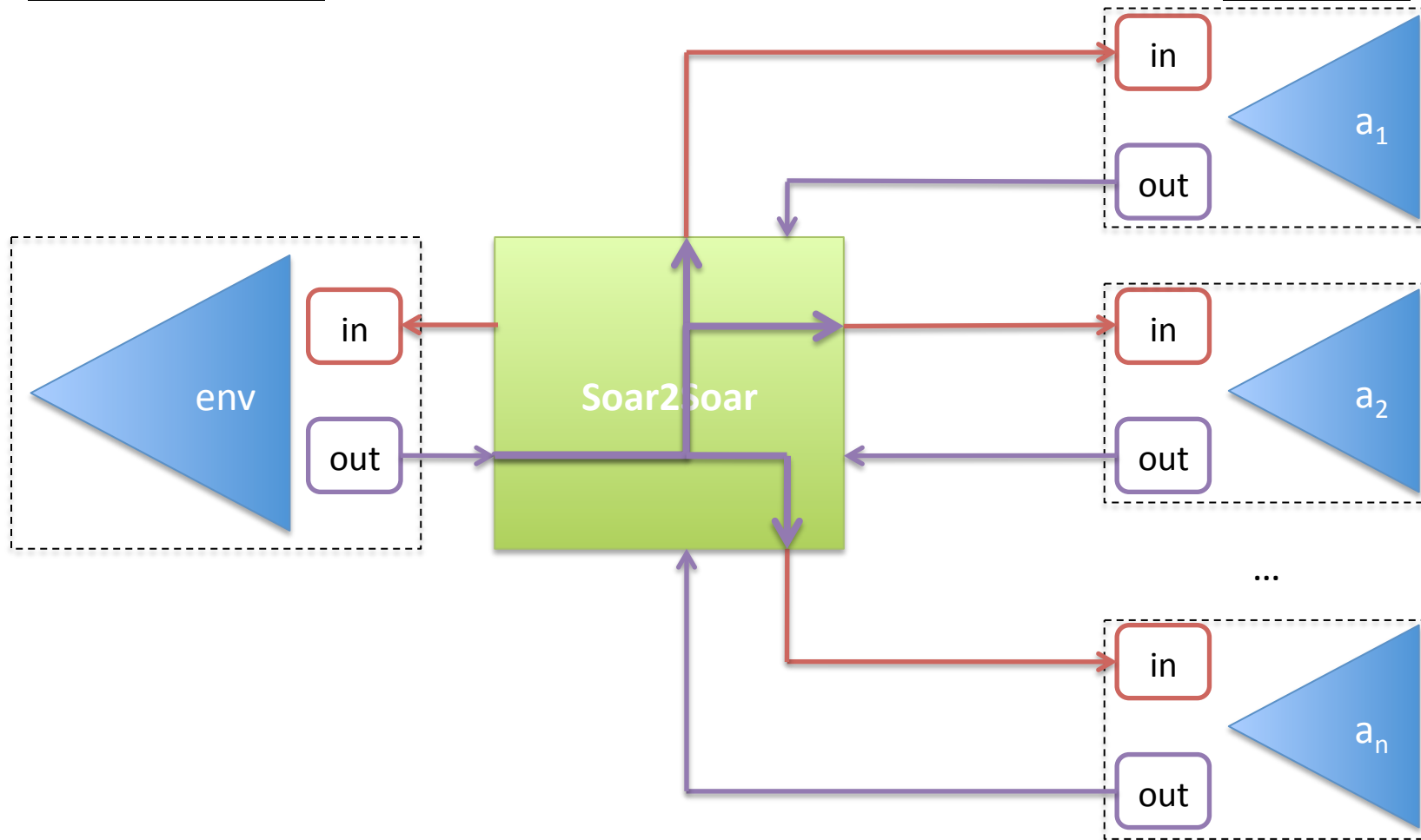
## Client(s)



# Soar2Soar: Env-to-Client

Environment

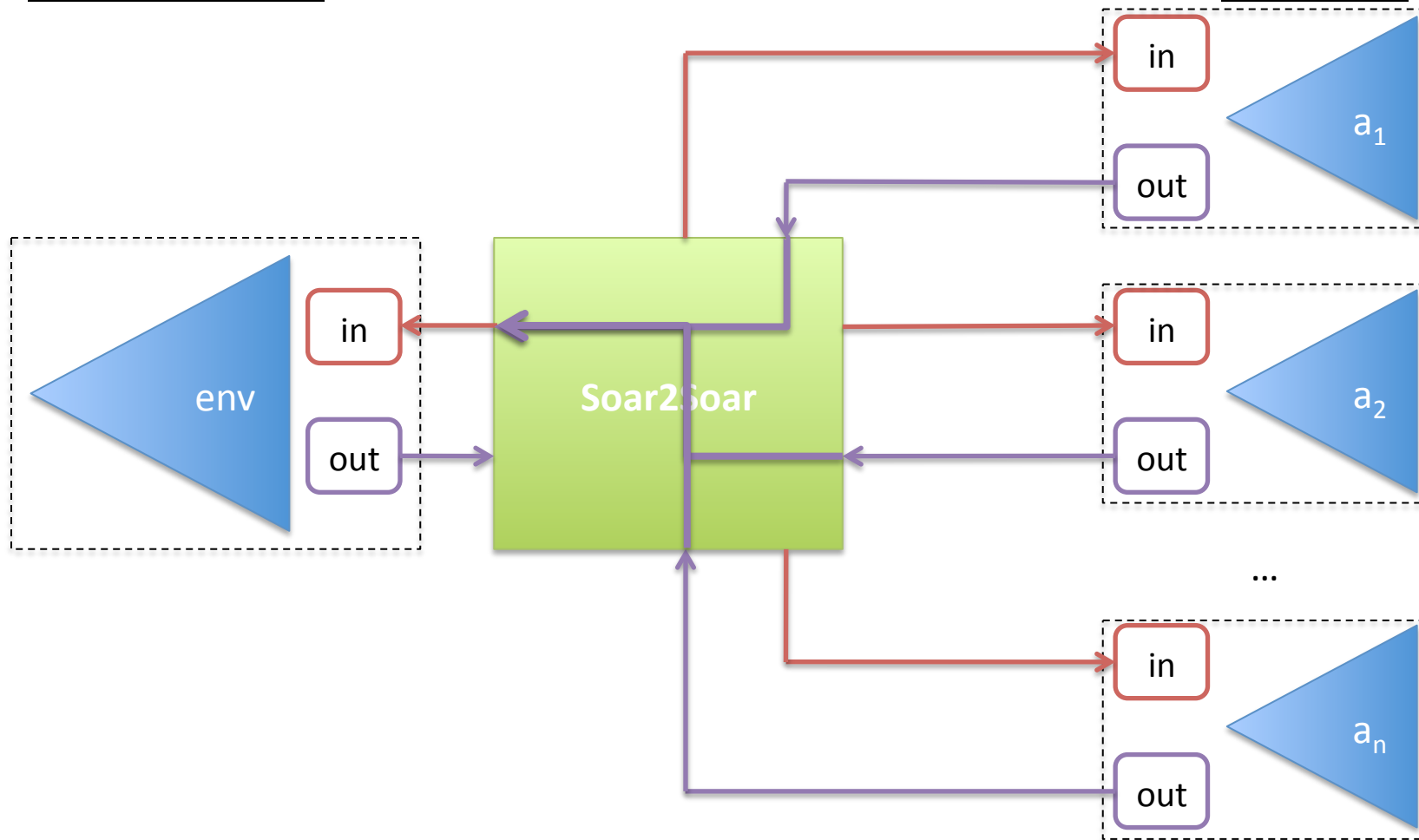
Client(s)



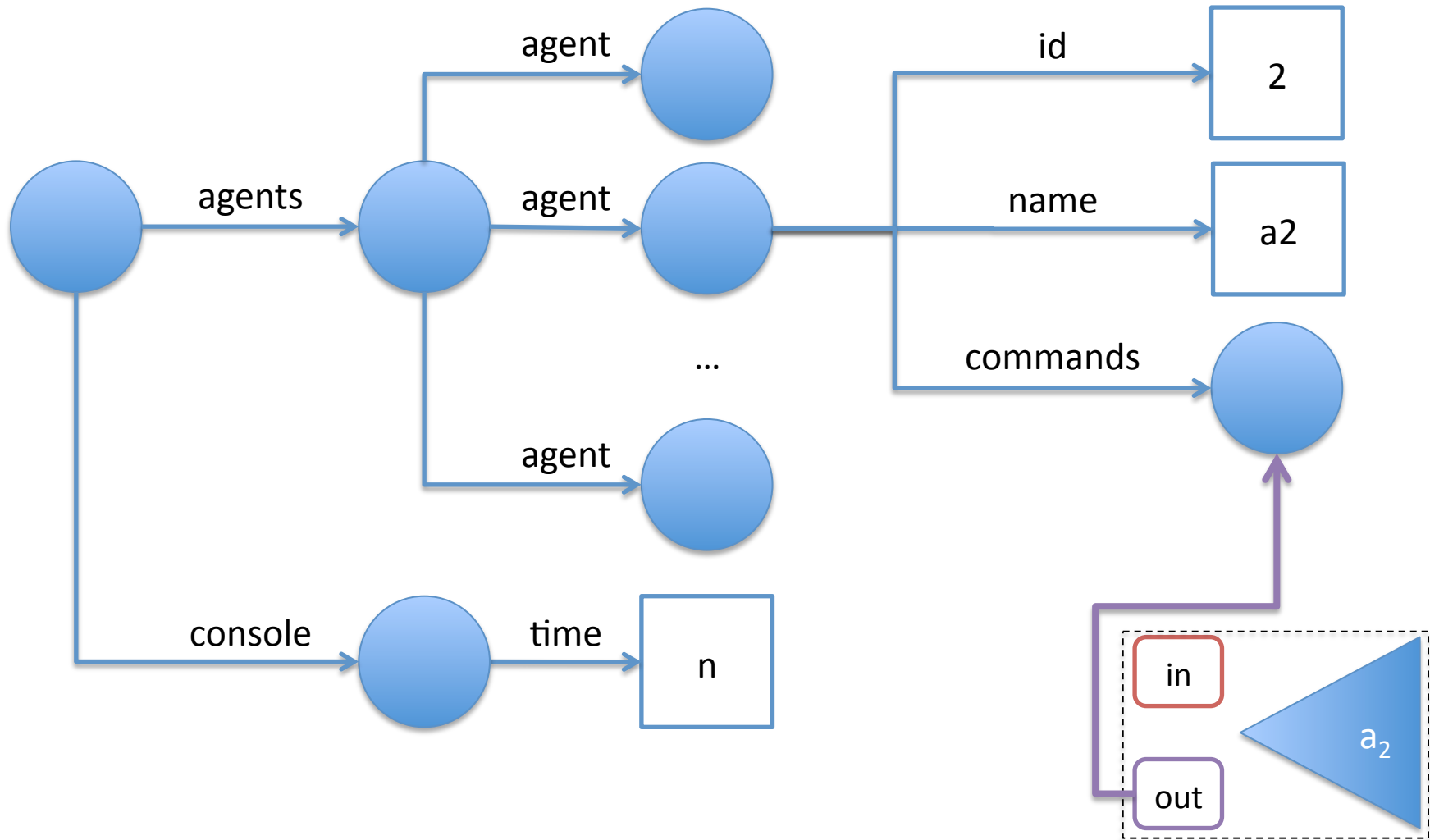
# Soar2Soar: Client-to-Env

Environment

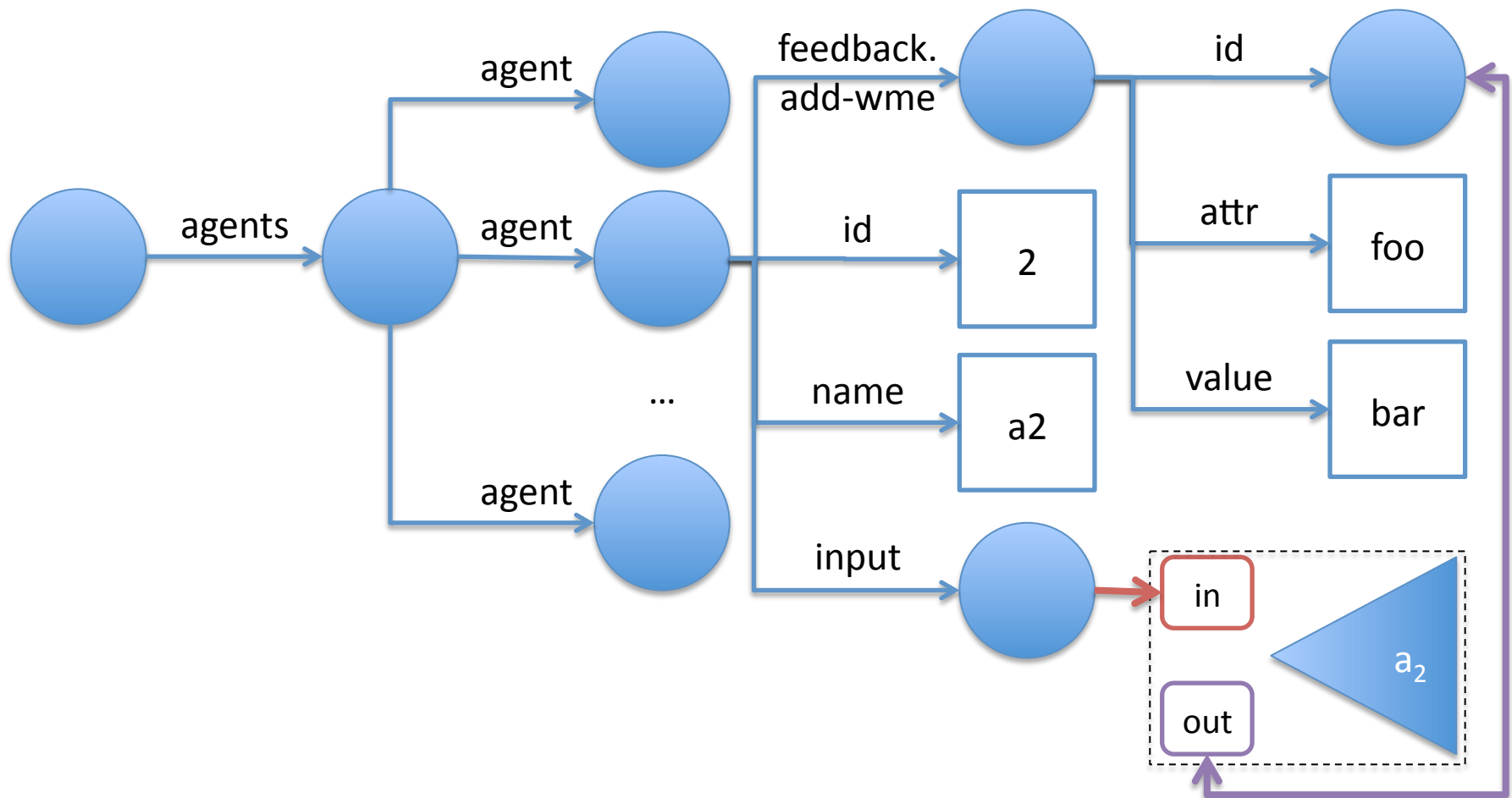
Client(s)



# Environment Input-Link



# Environment Output-Link





# Using Soar2Soar

No data declaration

Soar2Soar auto-magically maps outputs-to-inputs

```
$ soar2soar <n> [env source] [a1 source] [a2 source] ...  
> Once running, provides CLI for all agents and supports remote access
```

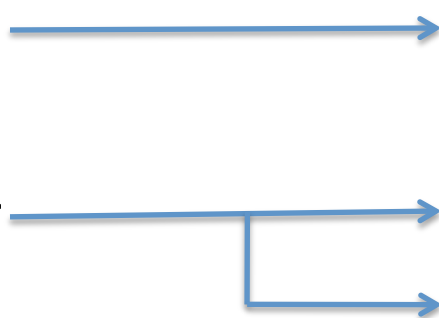
# Demo Agents

## Clients

- Blocks-World
- World-Mover

## Environments

- Blocks-World
- World
- World-Occupants



# Blocks-World

## Environment

15 rules

Arbitrary number of independent blocks-world instances

- 4 blocks
- Same start/end states
- Provides completion time for each agent

## Client

14 rules

Random movement

# Blocks-World Data



# World

## Environment

36 rules

Arbitrarily-sized grid world

- Set x/y limits in initialization rule

Single “move” command (NSEW)

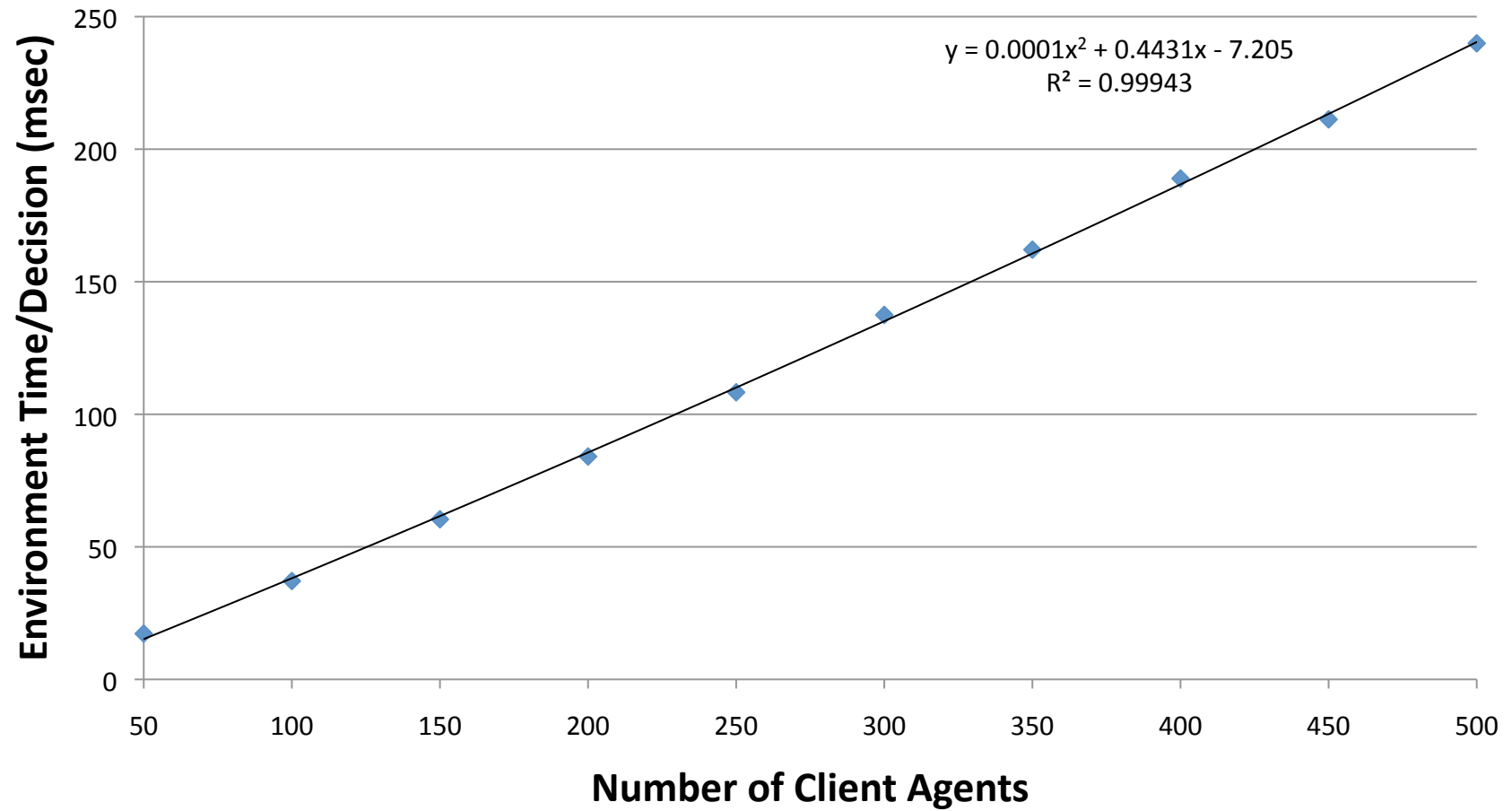
- Multiple agents allowed in each cell
- Failure only at borders

## Client

11 rules

Learns boundaries of the world

# World Data



# World-Occupants

Major change: environment supplies each client with features of occupants in neighboring grid cells

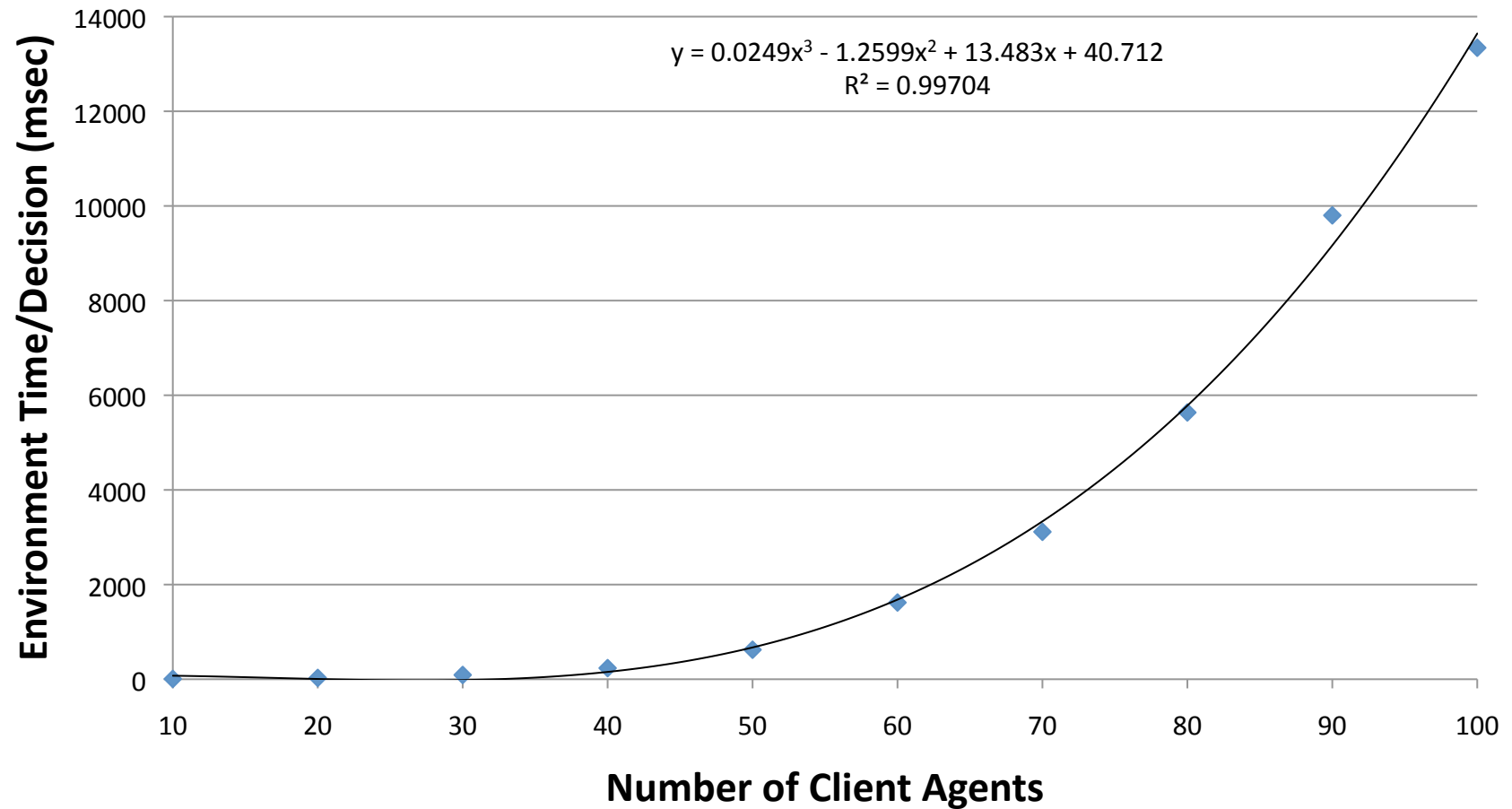
```

sp {elaborations*elaborate*occupants
  (state <s> ^name world
    ^io.output-link.agents <as>
    ^grid <grid>)
  (<as> ^agent <a1> {<> <a1> <a2>})
  (<a1> ^input <a1-il>)
  (<a2> ^input <a2-il>)
  (<a1-il> ^here <a1-h>
    ^<a1-d> <a1-relative>)
  (<a2-il> ^here <a2-h>
    ^race <a2-race>
    ^age <a2-age>)
  (<a1-h> ^x <a1-x>
    ^y <a1-y>)
  (<a2-h> ^x <a2-x>
    ^y <a2-y>)
  (<grid> ^spot <a1-s> <a2-s>)
  (<a1-s> ^x <a1-x>
    ^y <a1-y>
    ^{<< up down left right >> <a1-d>} <a2-s>)
  (<a2-s> ^x <a2-x>
    ^y <a2-y>)

```



# World-Occupants Data



# Evaluation

## Nuggets

- Fast environment prototyping using only Soar rules [in 9.3.0]
- Preliminary support for multi-agent models

## Coal

- Declarative representation makes inefficient operations very tempting
- Need to explore the degree to which parallelizing client decisions will improve run-time scaling