



Character AI Decision Making System for

FINAL FANTASY XV -EPISODE DUSCAE-

SQUARE ENIX CO., LTD.

Yoji Shirakami

Programmer in 2nd Business Division

Kousuke Namiki

AI Programmer in 2nd Business Division

Youichiro Miyake

Lead AI Researcher in Advanced Technology Division

Takanori Yokoyama

Programmer in 2nd Business Division

登壇者紹介



Yoji Shirakami

Programmer
2nd Business Division

- ◆ Primarily responsible for
 - Building AI Graph Editor
 - Creating Buddy AI



Youichiro
Miyake

Lead AI Researcher
Advanced Technology Division

- ◆ Primarily responsible for
 - Technical design of AI system
 - Deciding AI specifications



Kousuke Namiki

AI Programmer
2nd Business Division

- ◆ Primarily responsible for
 - Building AI common system
 - Creating Enemy AI

Audience of interest / required Skill



- Engineers who want to improve character's decision making system.
- Game designers who want to realize smarter and stronger character's behavior control.
- Game developers who want to learn promising AI technologies.
- Researchers who have interest in new game AI techniques.
- Any one with limited AI knowledge can understand the lecture.
- The lecture includes tool engineering techniques in tools section.

Knowledge you can take away



- Overview of current decision making technologies.
- Original decision making system in FINAL FANTASY XV -EPISODE DUSCAE-
- Know-how to implement and design graph-based AI logic tool.
- Case study of applying these AI technologies to characters in-game.

Agenda



- Introduction to the lecture 3min.
- Introduction to decision-making system (Miyake) 5min.
- Luminous AI Technical design (Miyake) 5min.
- AI Graph Editor Implementation (Shirakami) 15min.
- Case study in-game (Namiki, Shirakami) 25min.
- Session summary 2min.

Agenda



- Introduction to the lecture 3min.
- Introduction to decision-making system (Miyake) 5min.
- Luminous AI Technical design (Miyake) 5min.
- AI Graph Editor Implementation (Shirakami) 15min.
- Case study in-game (Namiki, Shirakami) 25min.
- Session summary 2min.

INTRODUCTION

Introduction



- FFXV -EPISODE DUSCAE- AI Overview
- Requirements for creation of Character AI

Introduction



- FFXV -EPISODE DUSCAE- AI Overview
- Requirements for creation of Character AI

FFXV -EPISODE DUSCAE- AI Overview



Characters and making movie

©2015 SQUARE ENIX CO., LTD. All Rights Reserved.

Introduction

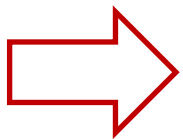


- FFXV -EPISODE DUSCAE- AI Overview
- Requirement for creation of Character AI

Requirement for creation of Character AI



- Rational and convincing action selection
- Various and real behaviors
- Seamless connection between event scene and character behaviors
- mass production of character AI variations



We need a tool to produce AI decision making module.

Agenda



- Introduction to the lecture 3min.
- Introduction to decision-making system (Miyake) 5min.
- Luminous AI Technical design (Miyake) 5min.
- AI Graph Editor Implementation (Shirakami) 15min.
- Case study in-game (Namiki, Shirakami) 25min.
- Session summary 2min.

What is decision making ?

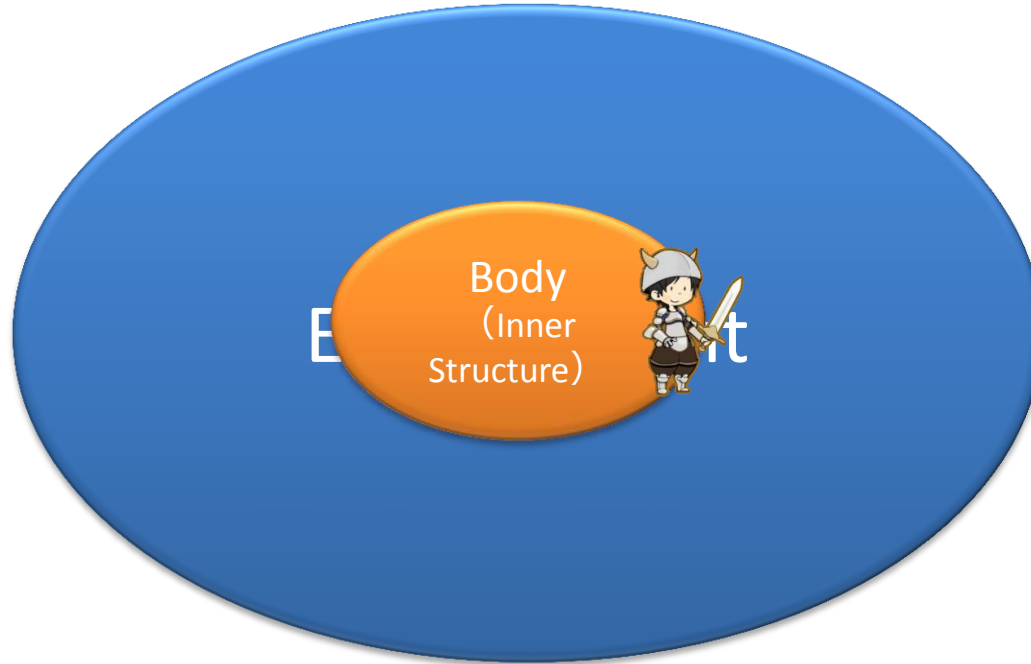
INTRODUCTION TO DECISION MAKING

What is intelligence ?

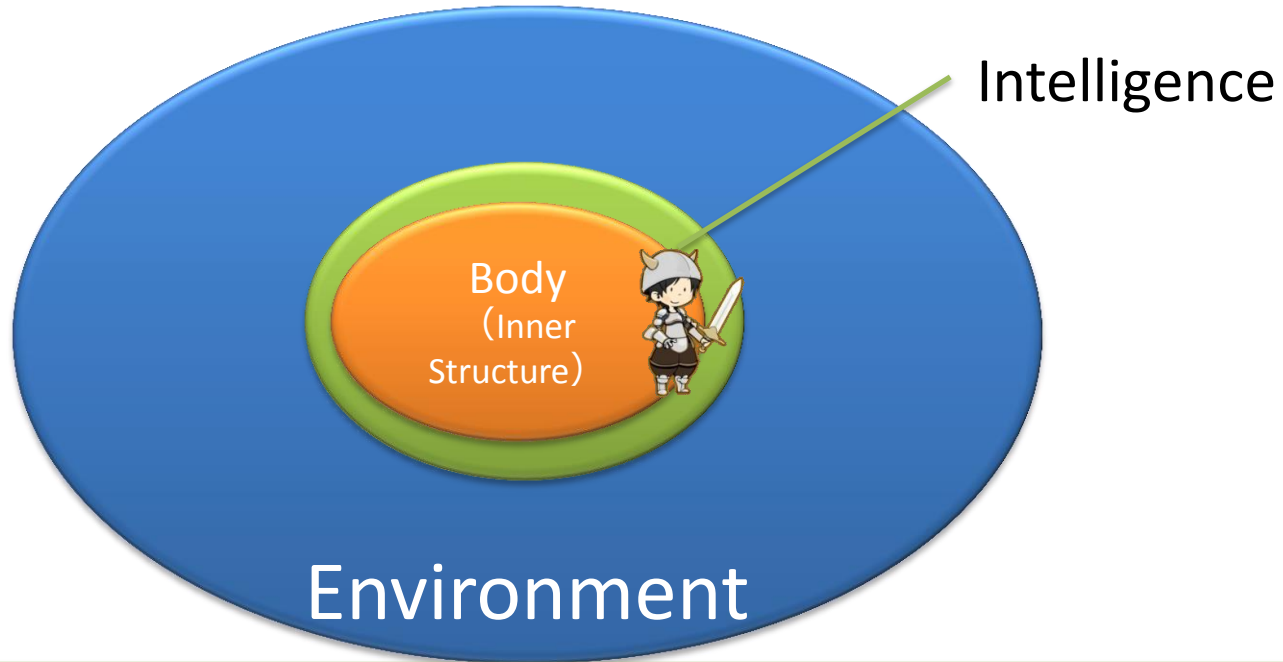


Environment

What is intelligence ?

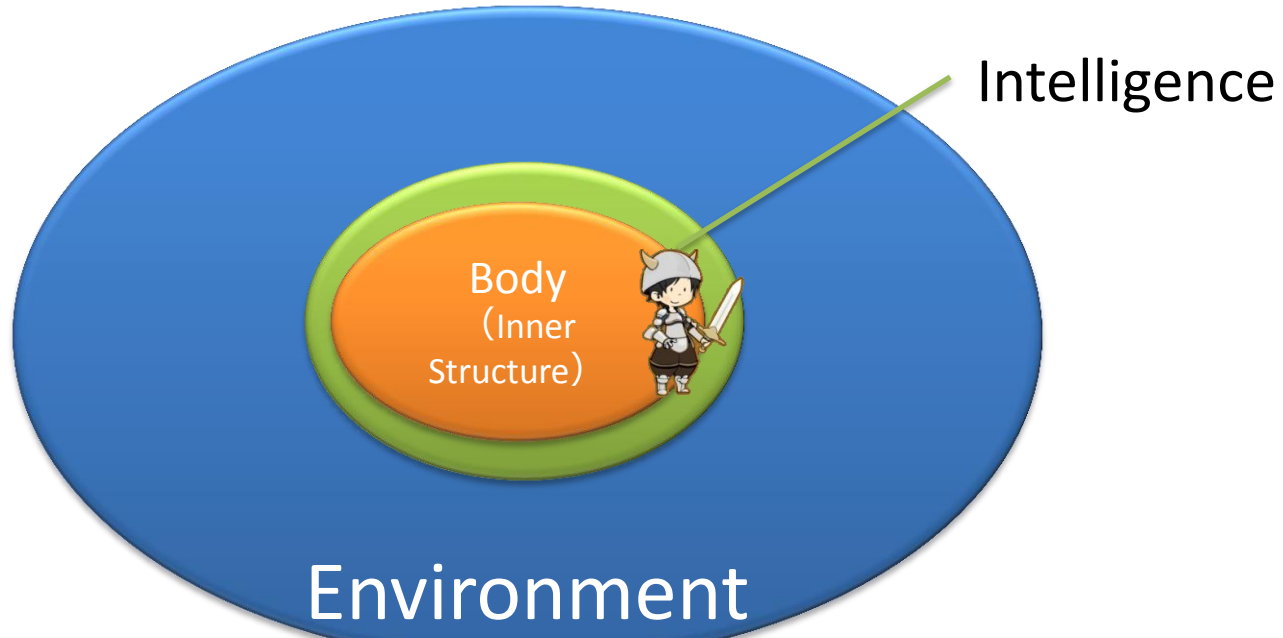


What is intelligence ?



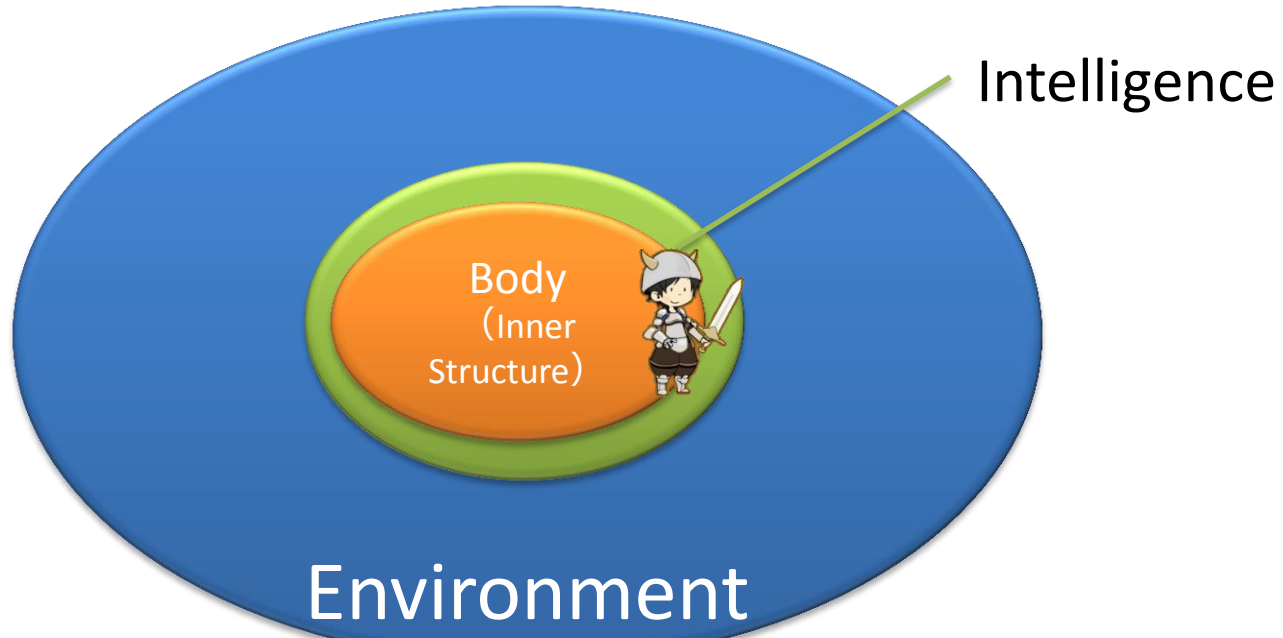
Intelligence = dynamically makes a body in harmony with environment.

What is intelligence ?



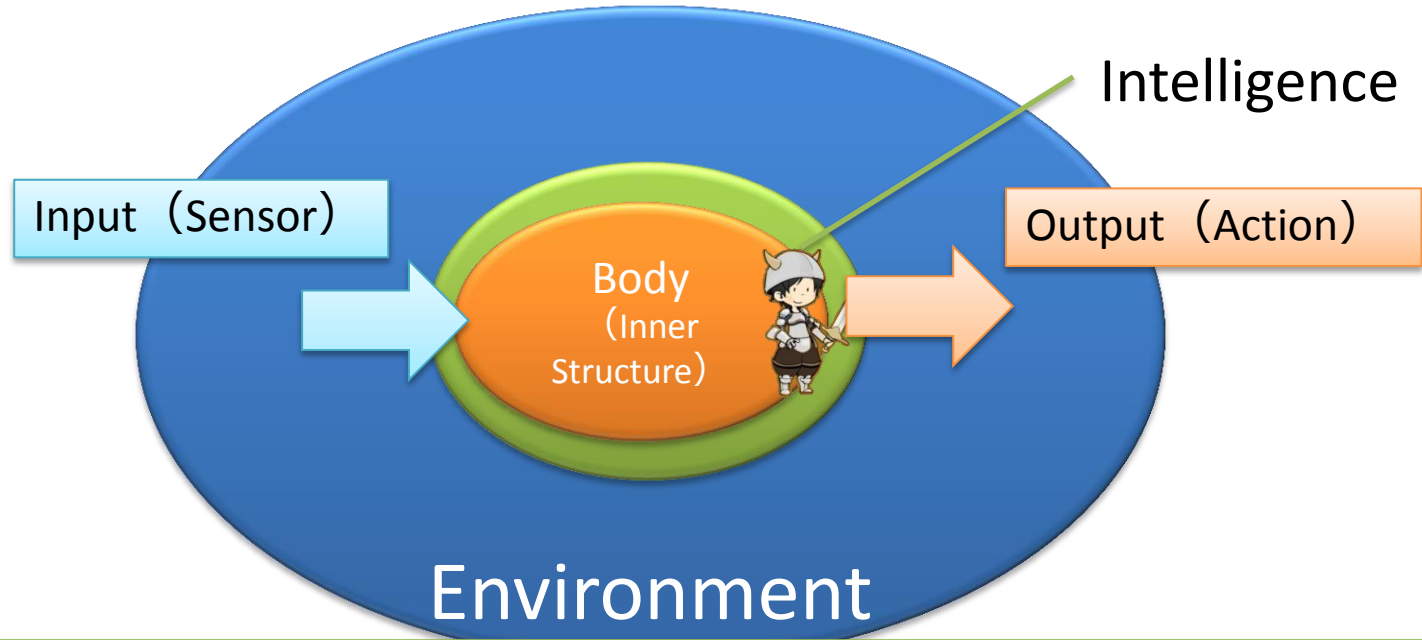
Artificial Intelligence = dynamically makes an Artificial Being in harmony with artificial environment.

What is intelligence ?



Artificial Intelligence = dynamically allows an Artificial Presence to act in the environment

What is intelligence ?



Artificial Intelligence = dynamically makes an AI's action in harmony with artificial environment.

Intelligence World

Sensor •
Body

Effector •
Body

Environment World

Intelligence World

Agent Architecture separates
Environment World and
Intelligence World

Sensor
Body

Effector •
Body

Thinking

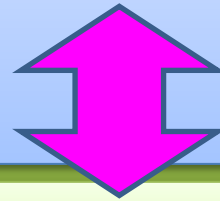
Memory

Sensor •
Body

Effector •
Body

Environment World

Thinking

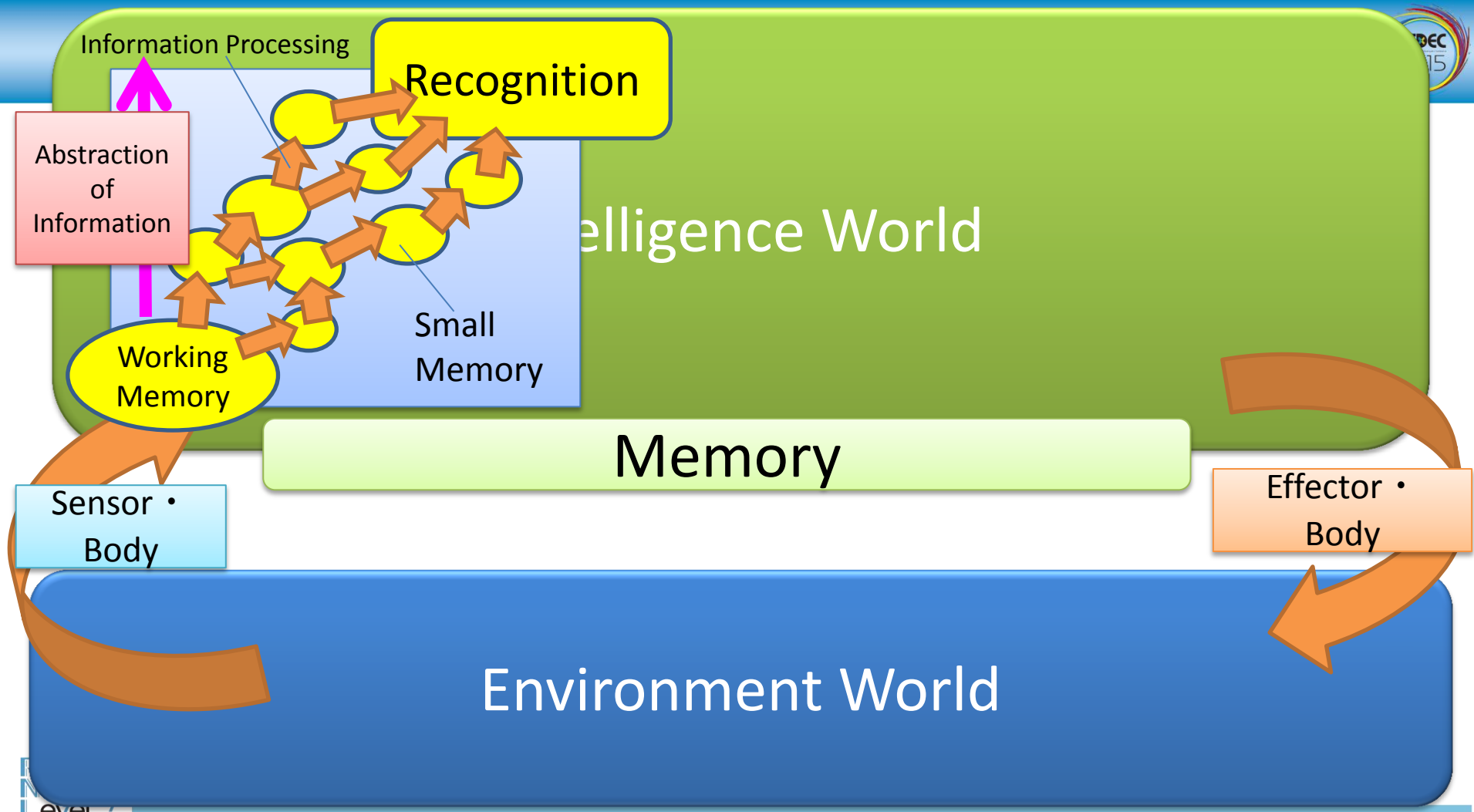


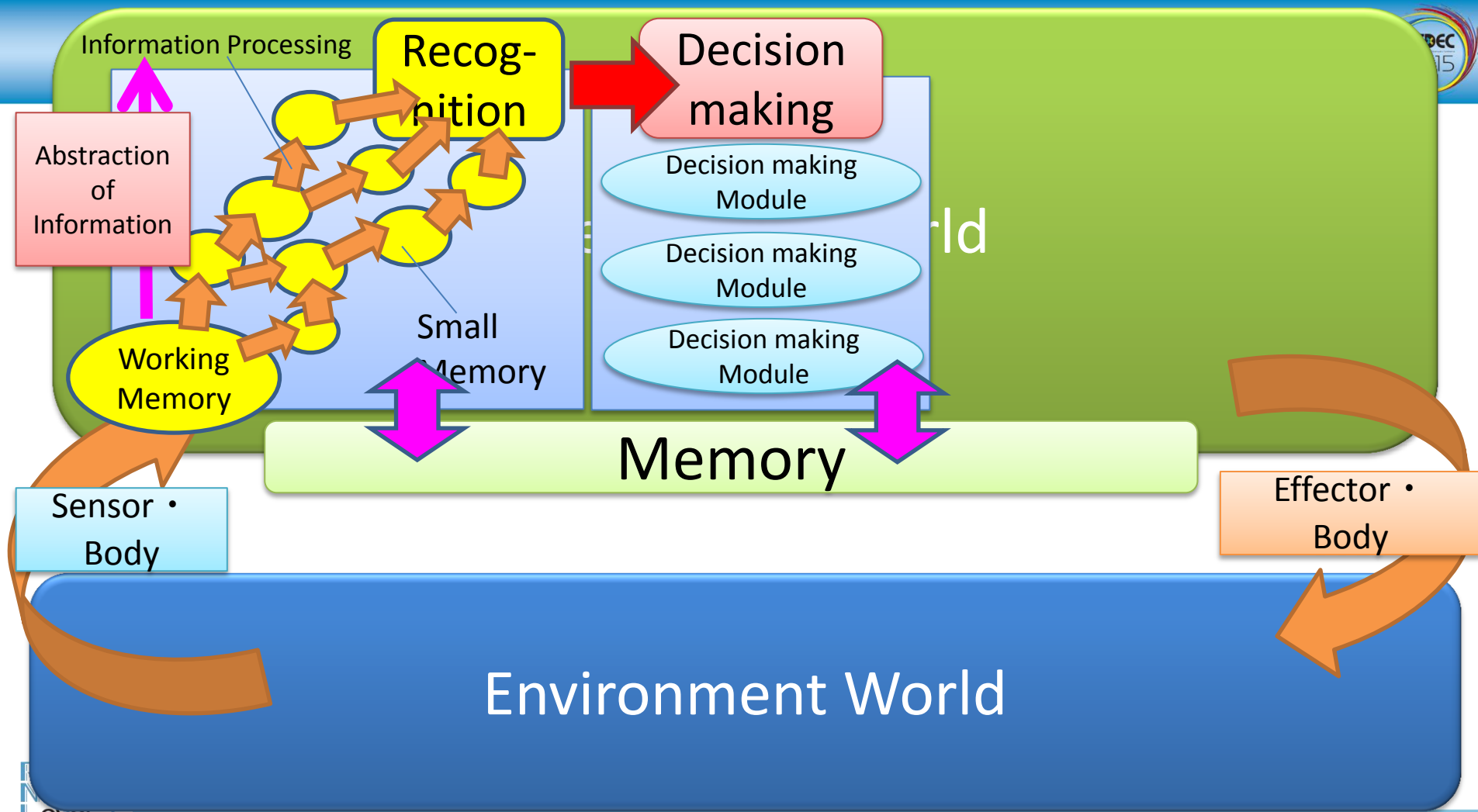
Memory

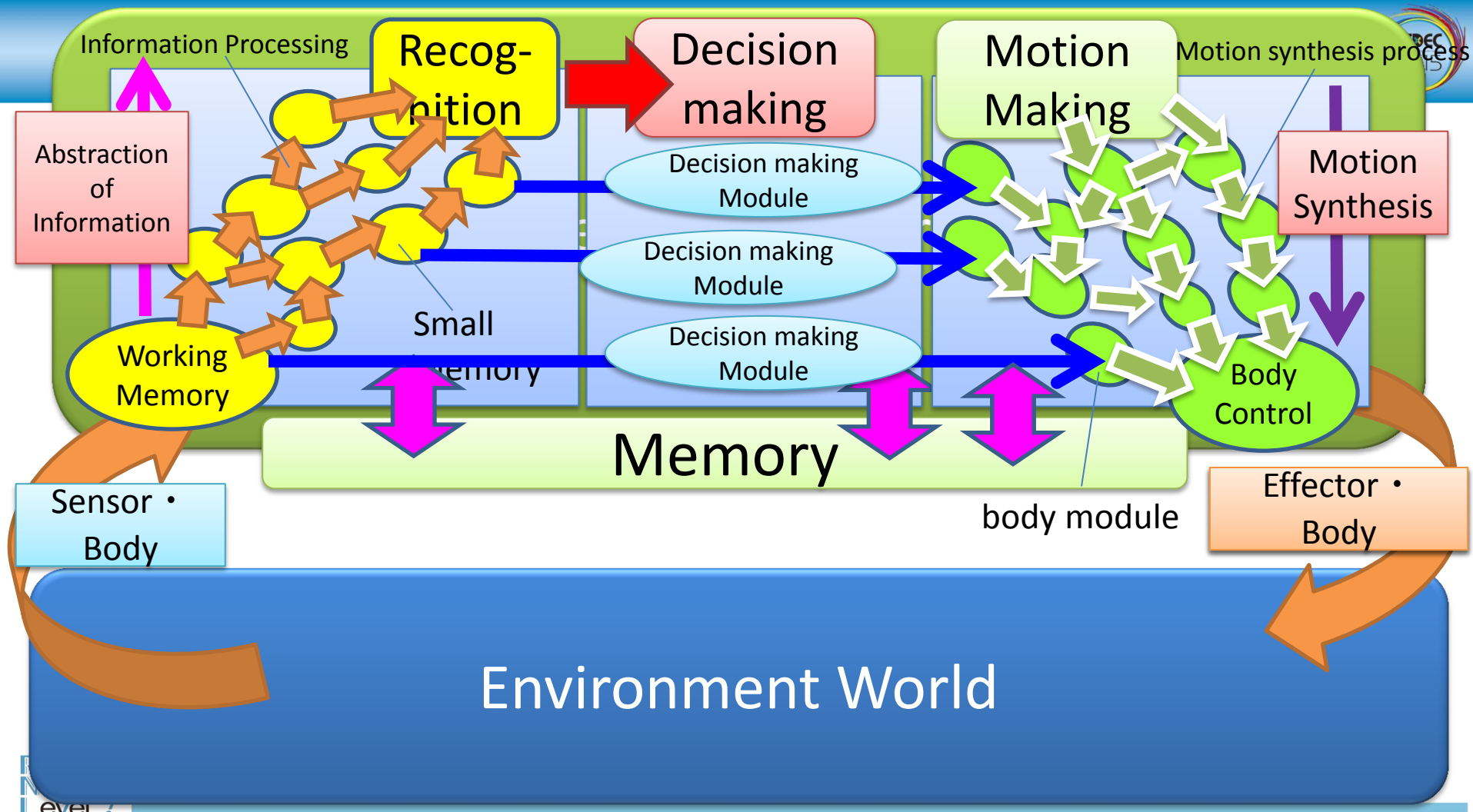
Sensor •
Body

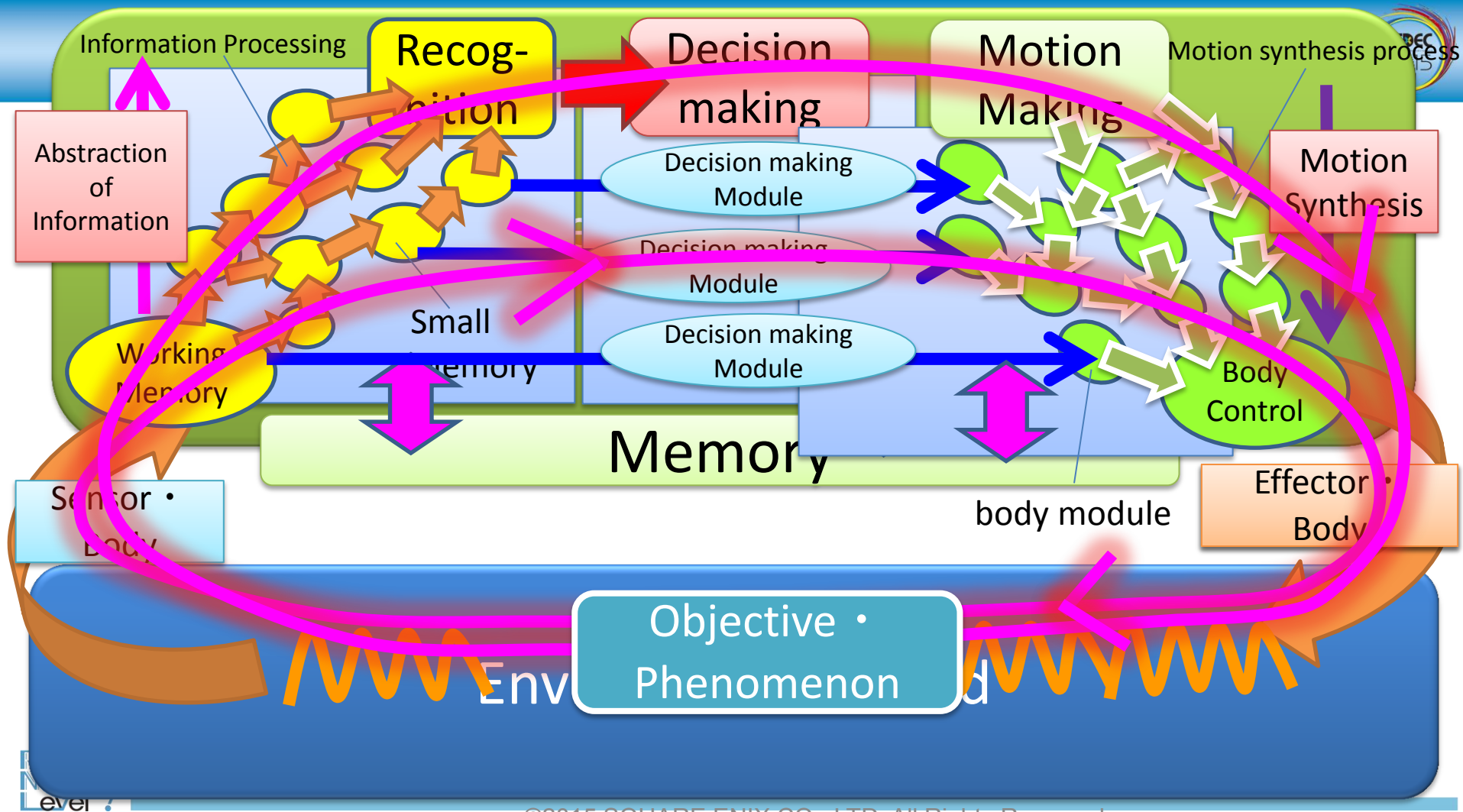
Effector •
Body

Environment World



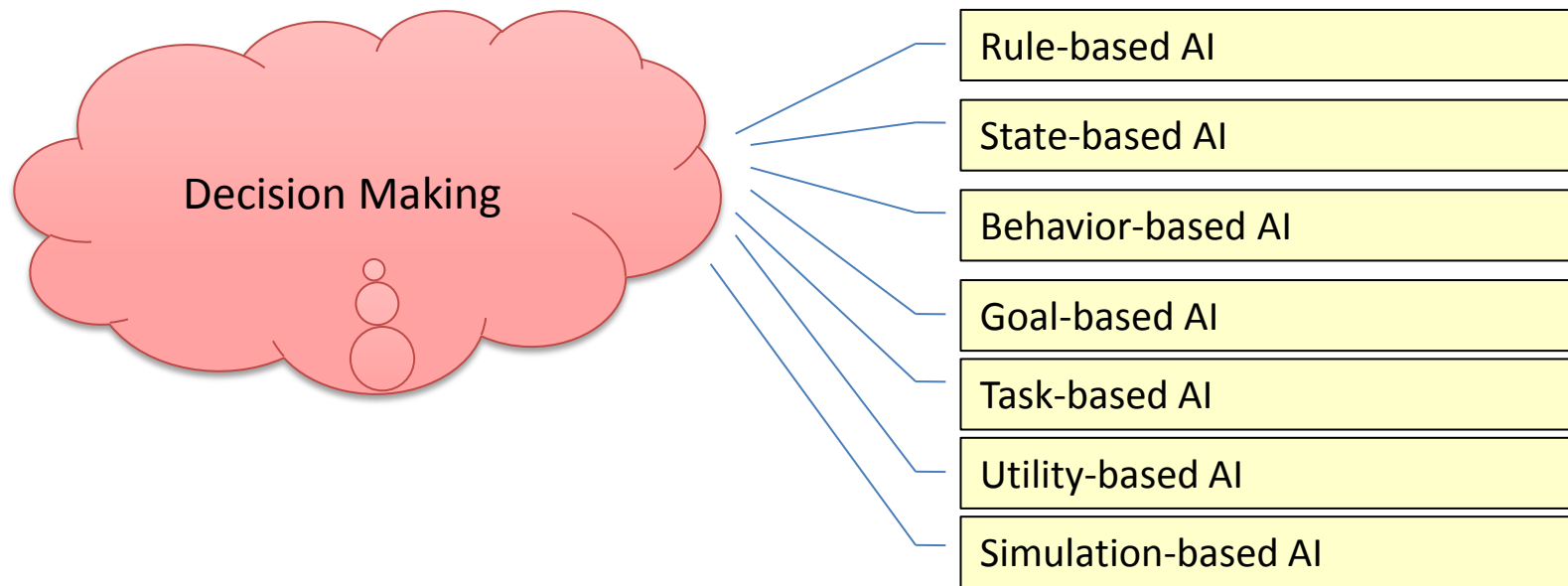






Decision Making Model

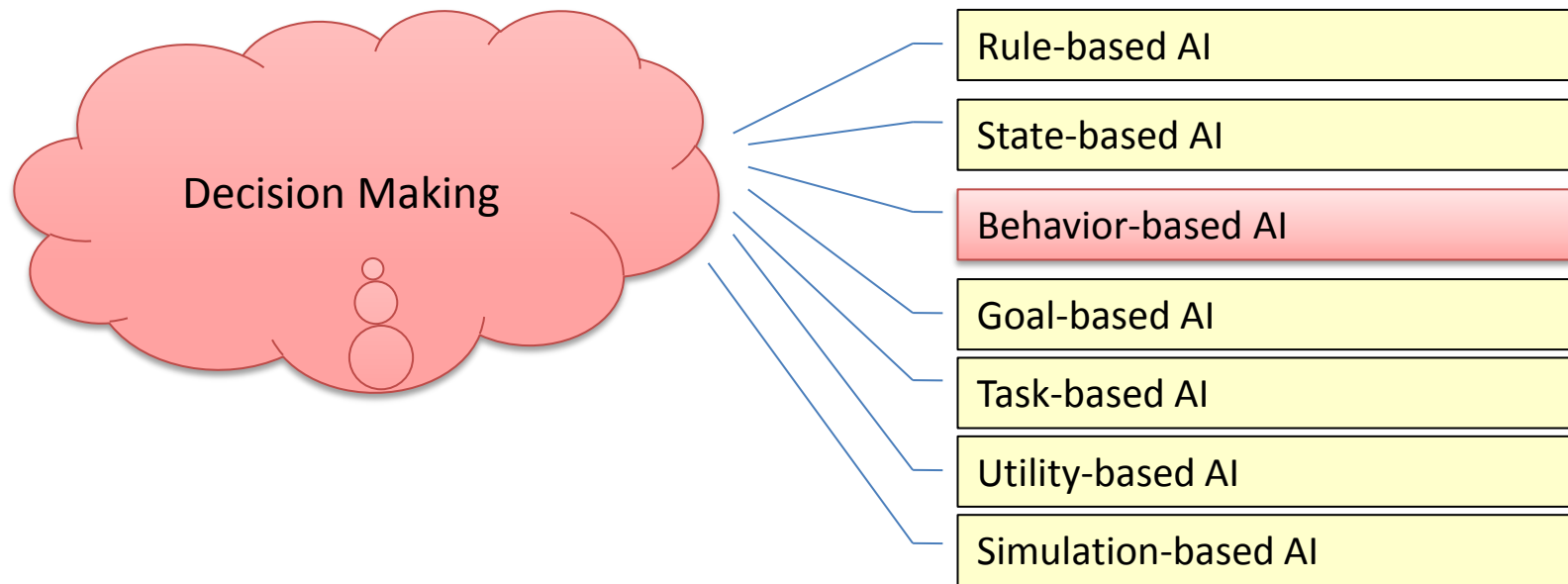
Decision Making is generally a very complex high-degree process. But for digital game there are 7 simple decision making algorithms.



「(something)-based AI」 means that an algorithm uses (something) as a unit.

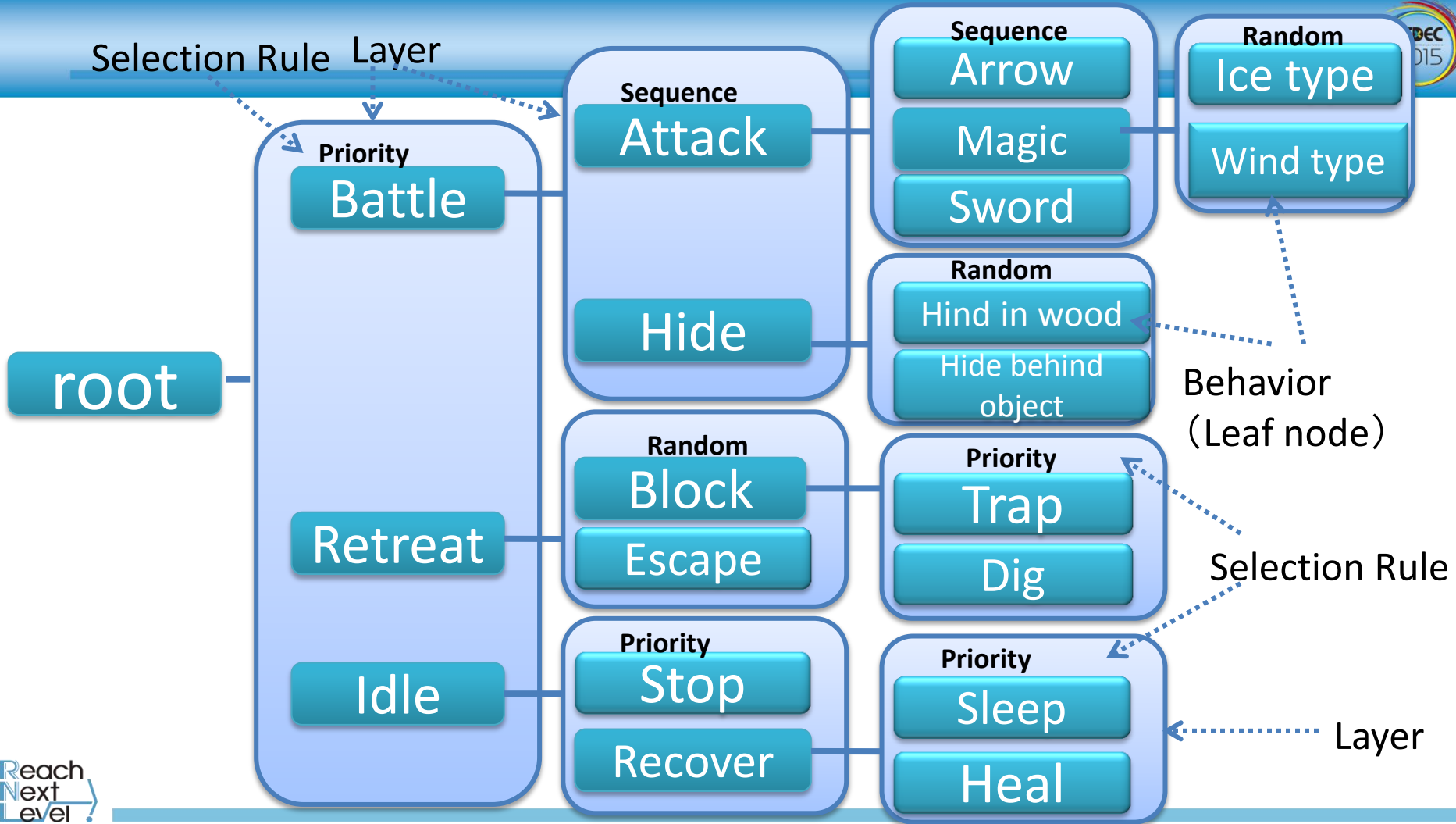
Decision Making Model

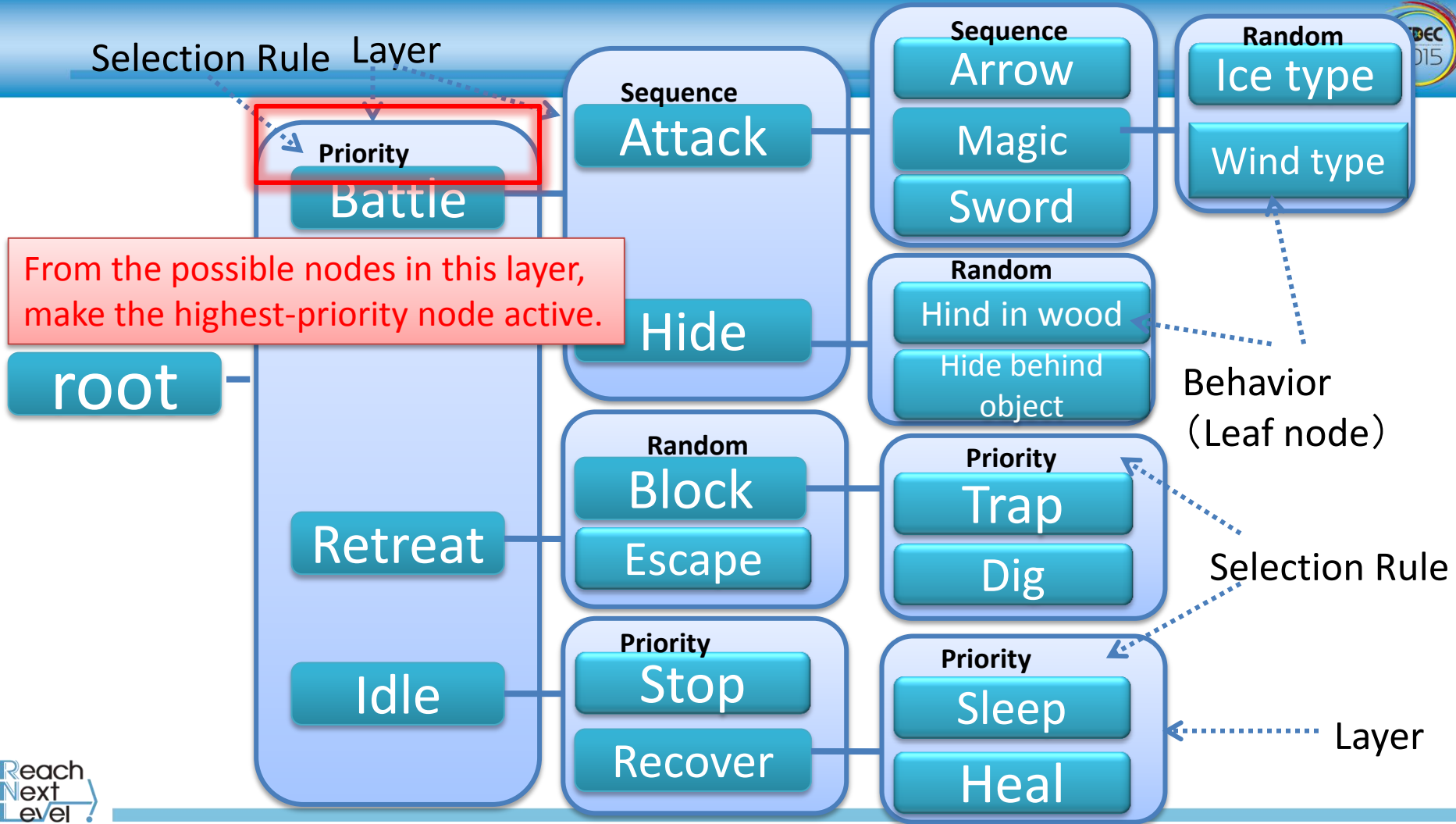
Decision Making is generally a very complex high-degree process. But for digital game there are 7 simple decision making algorithms.

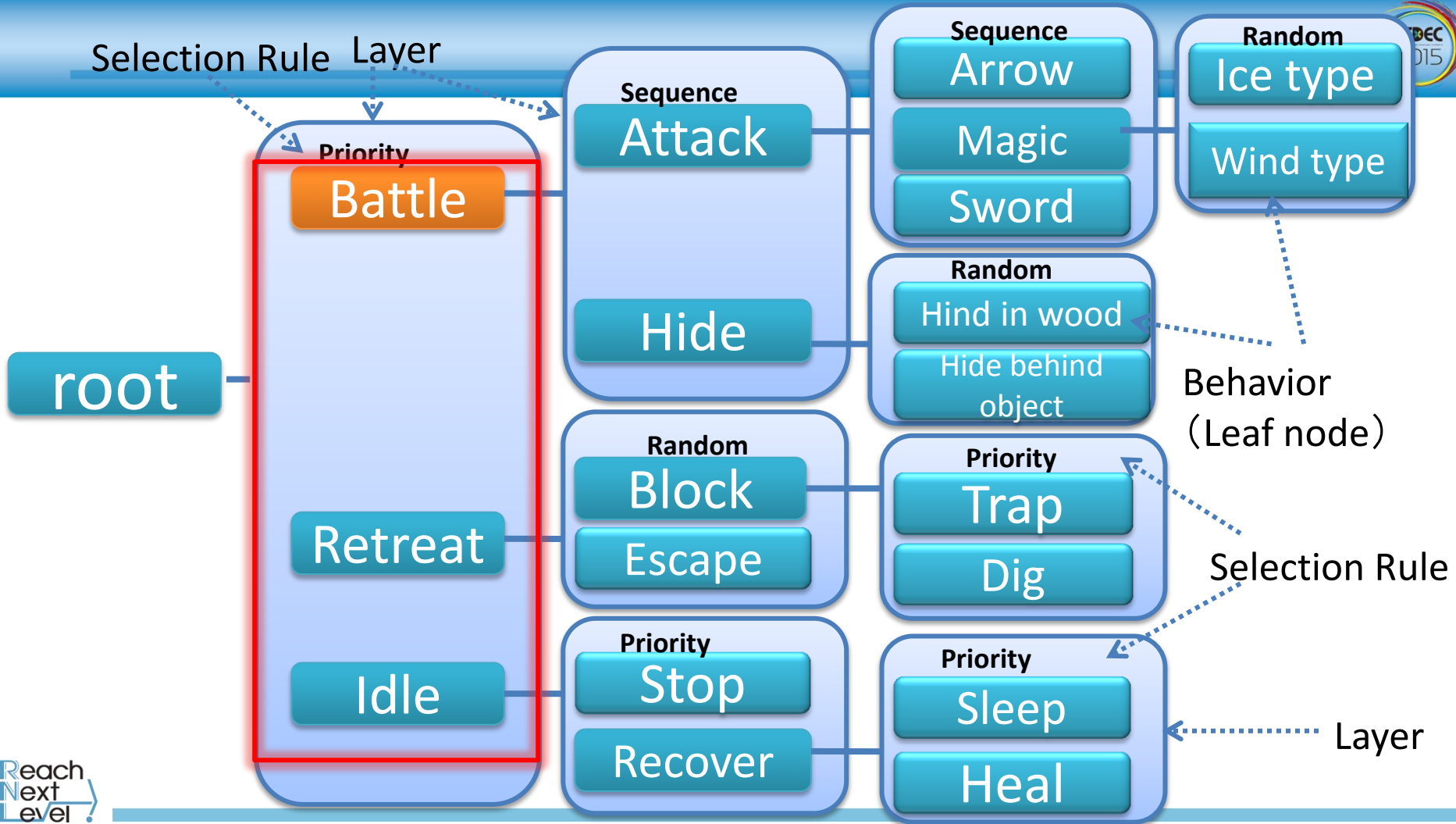


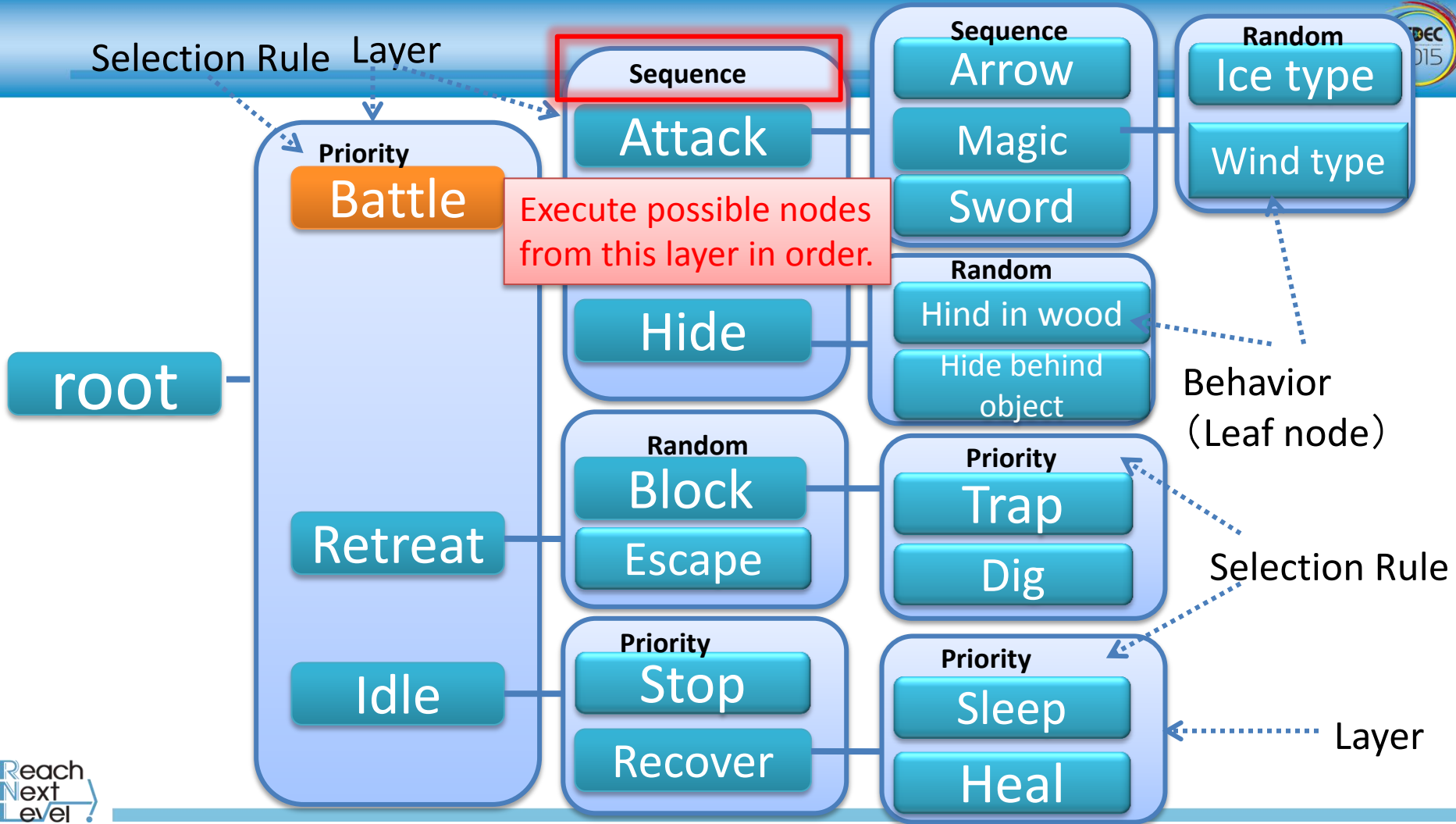
「(something)-based AI」 means that an algorithm uses (something) as a unit.

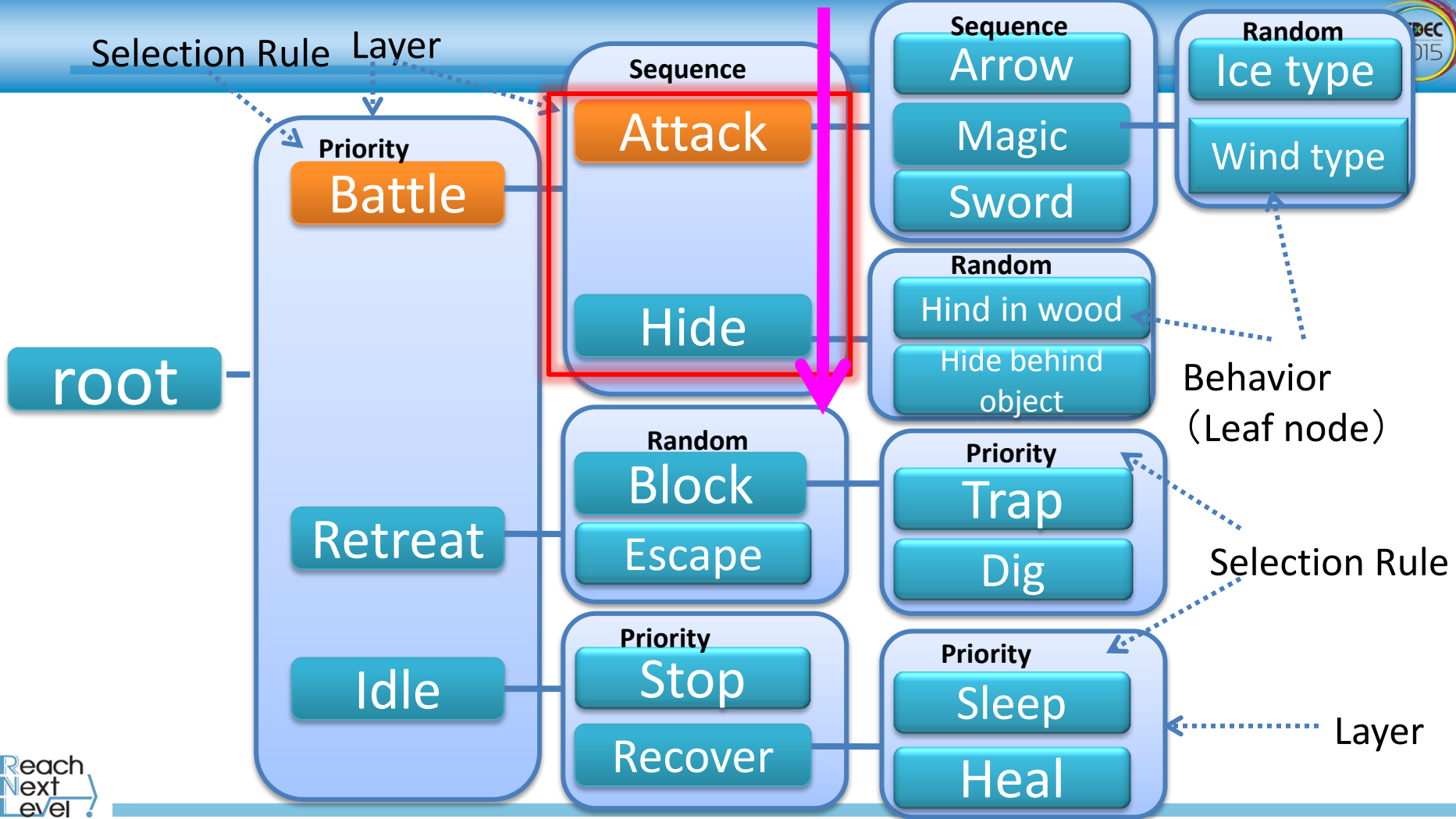
INTRODUCTION TO BEHAVIOR TREES

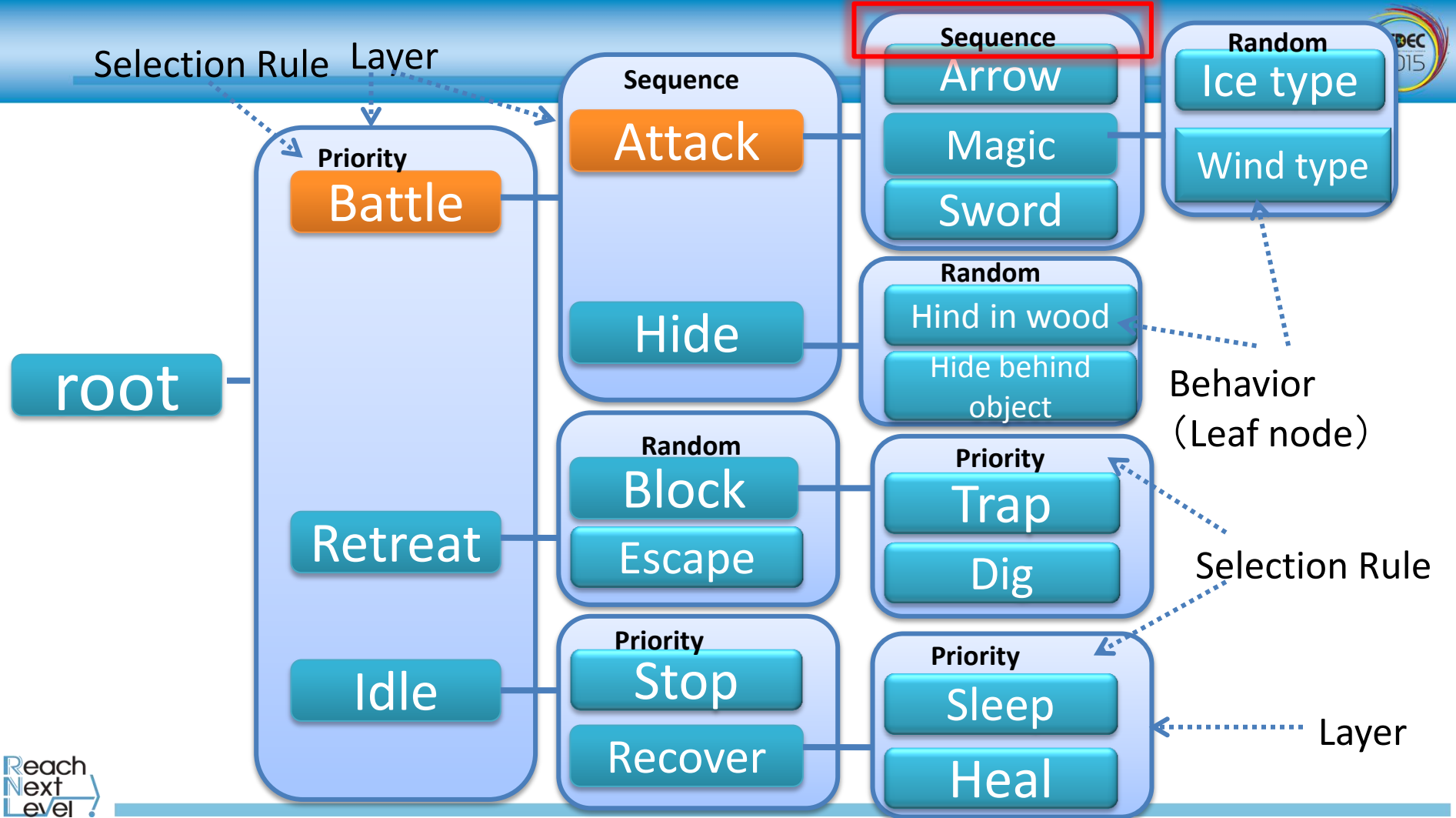




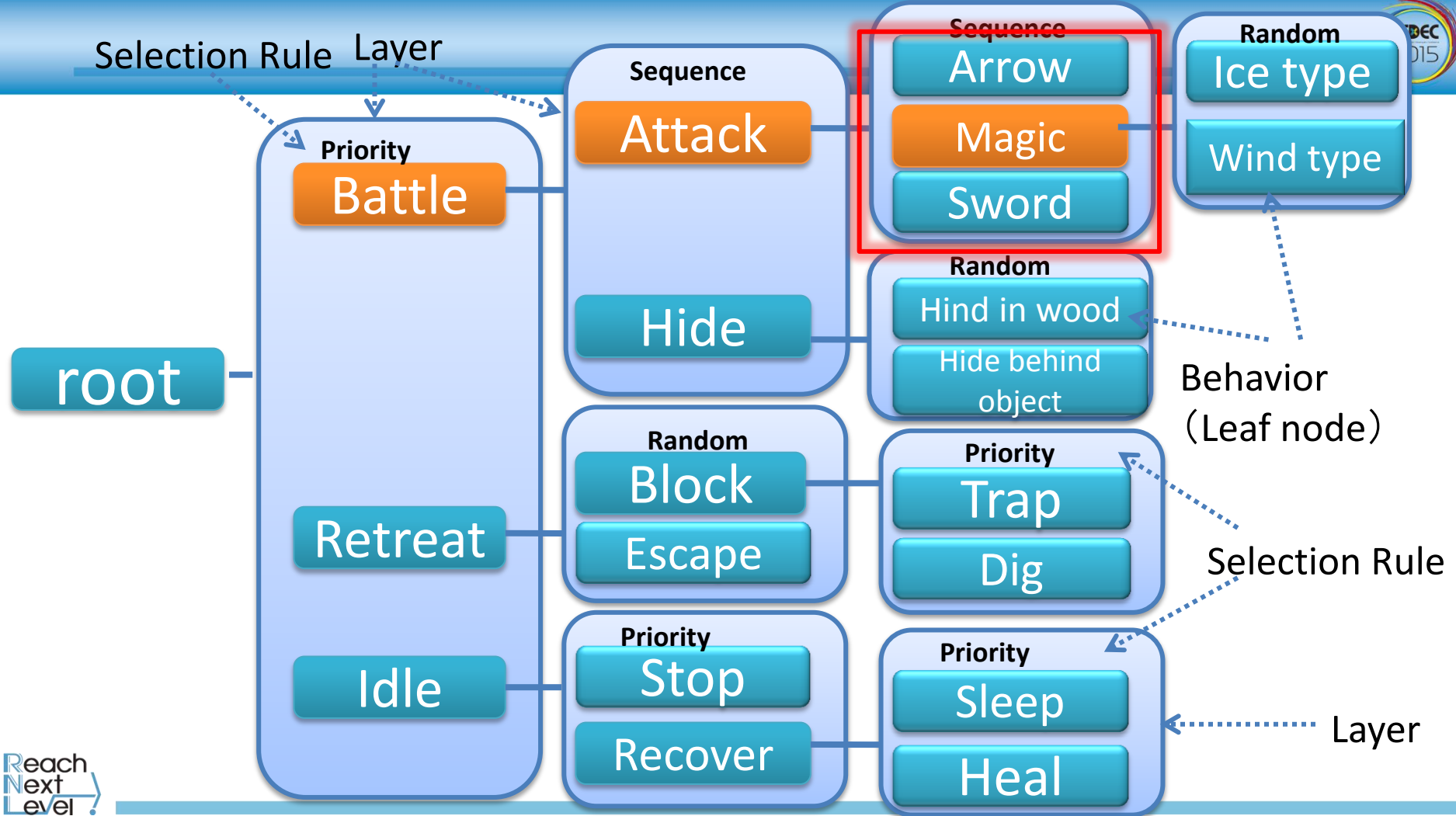


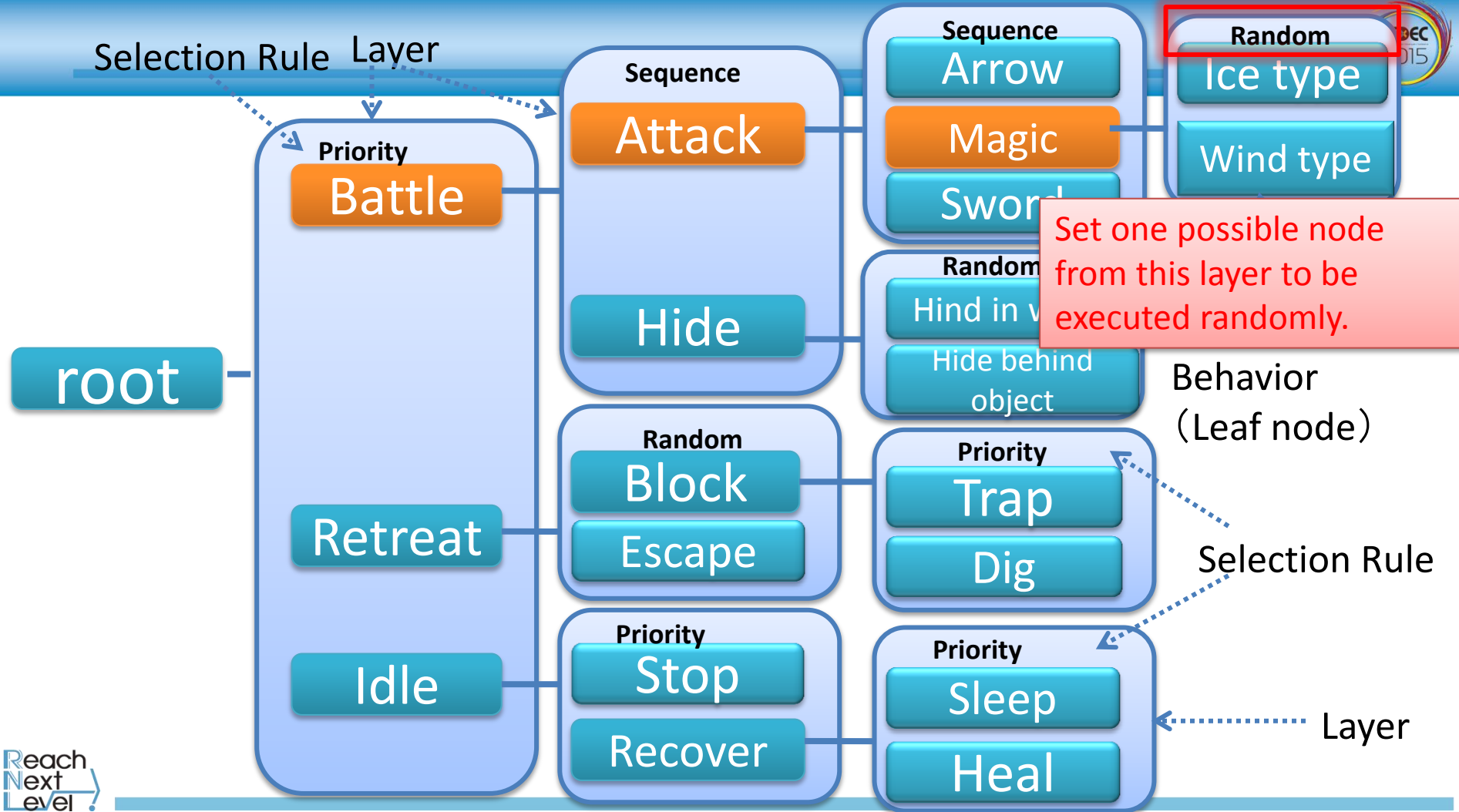


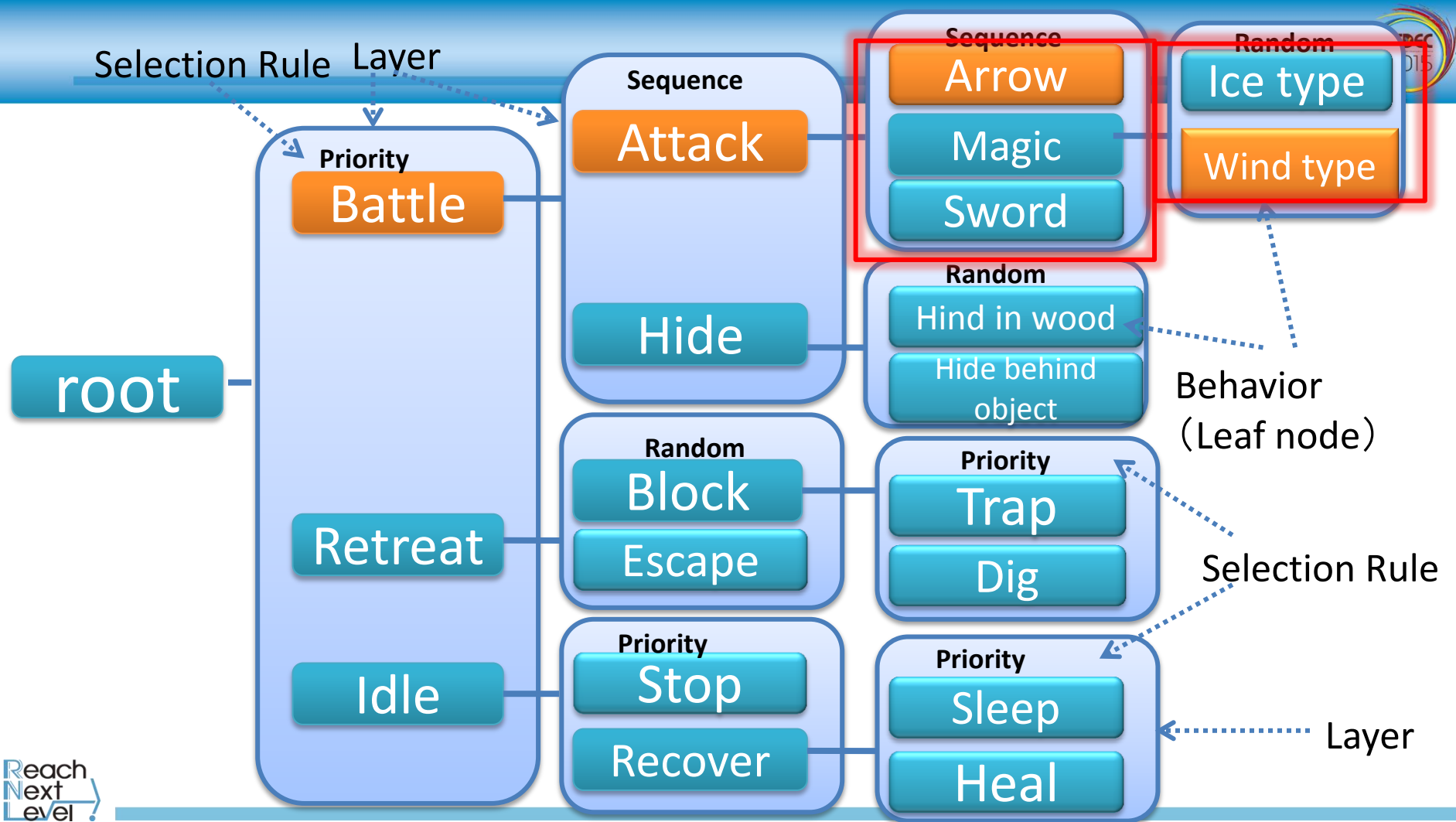


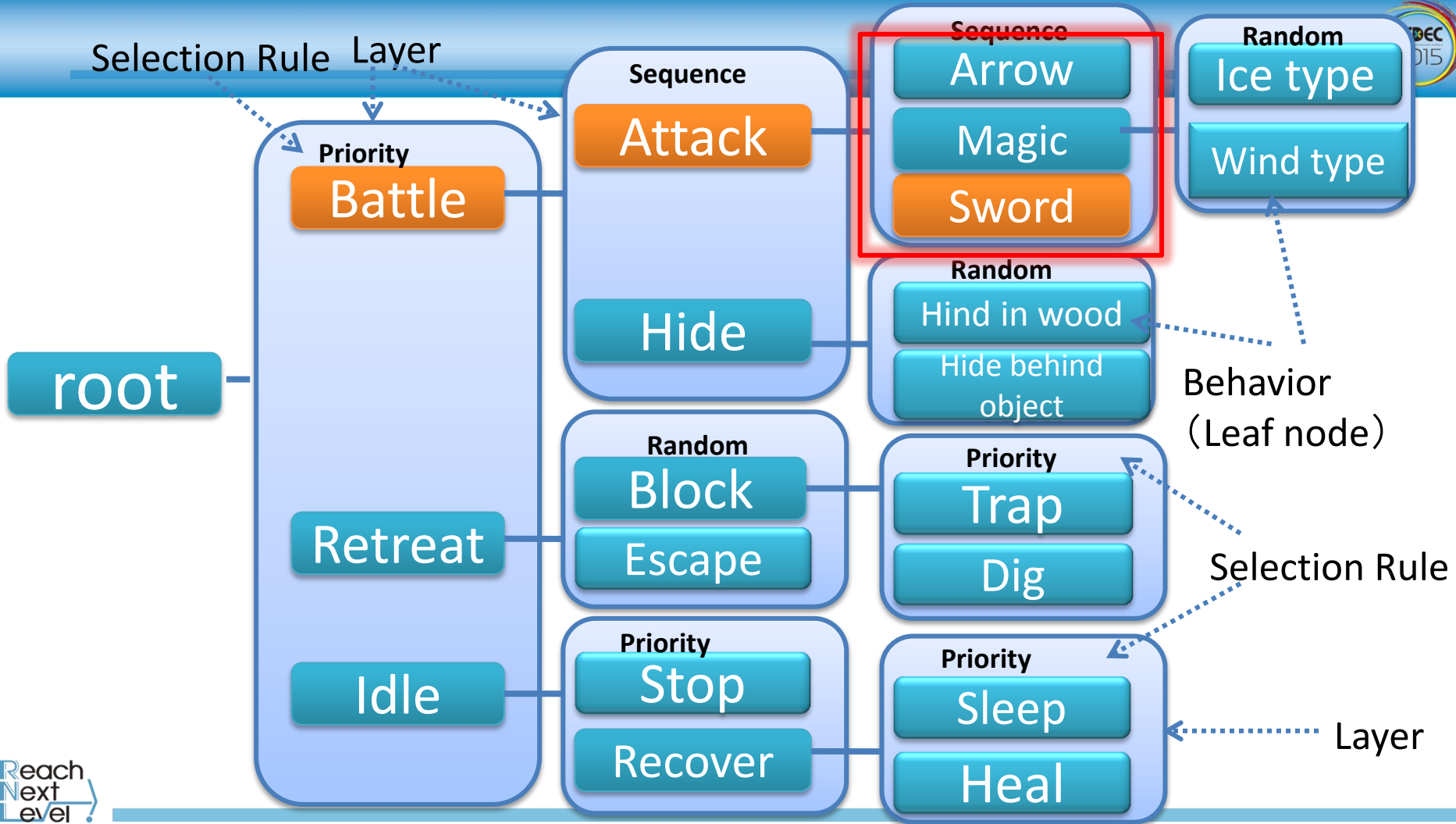


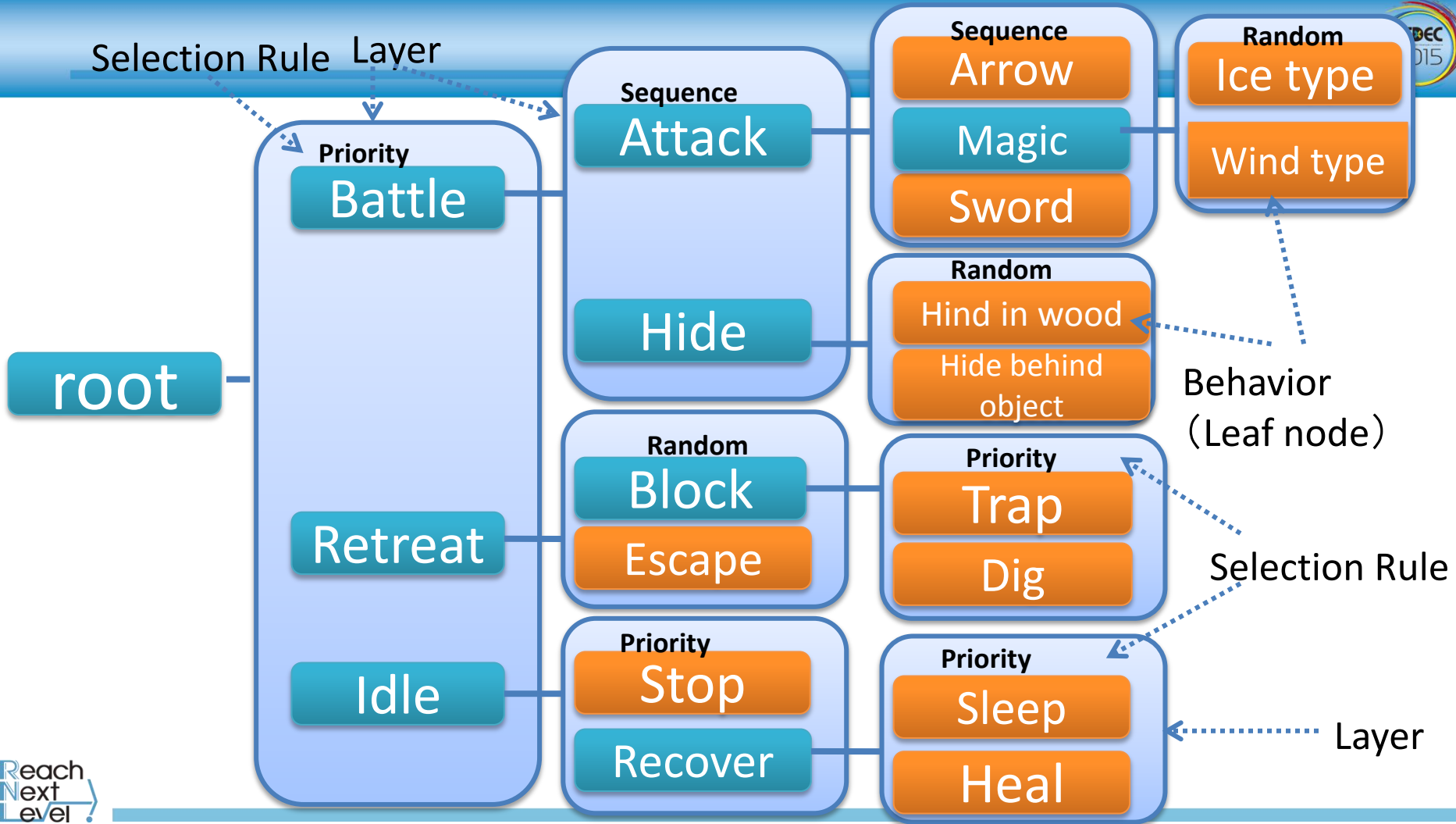






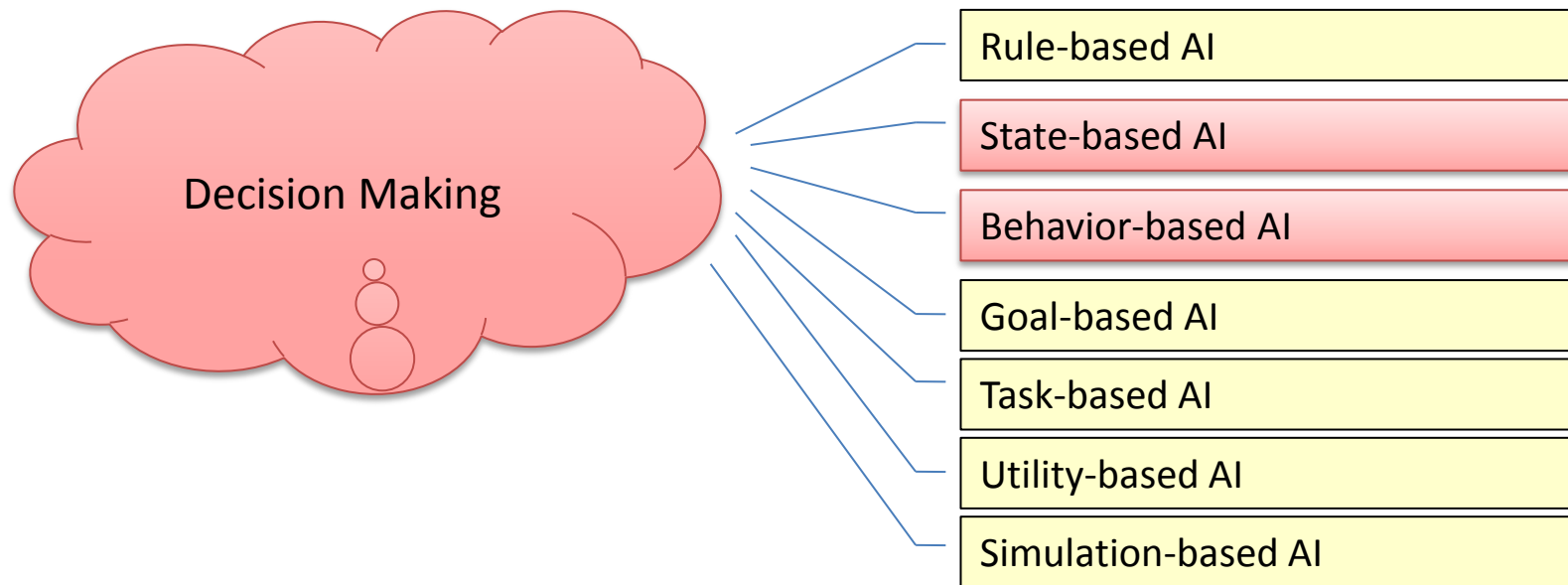






Decision Making Model

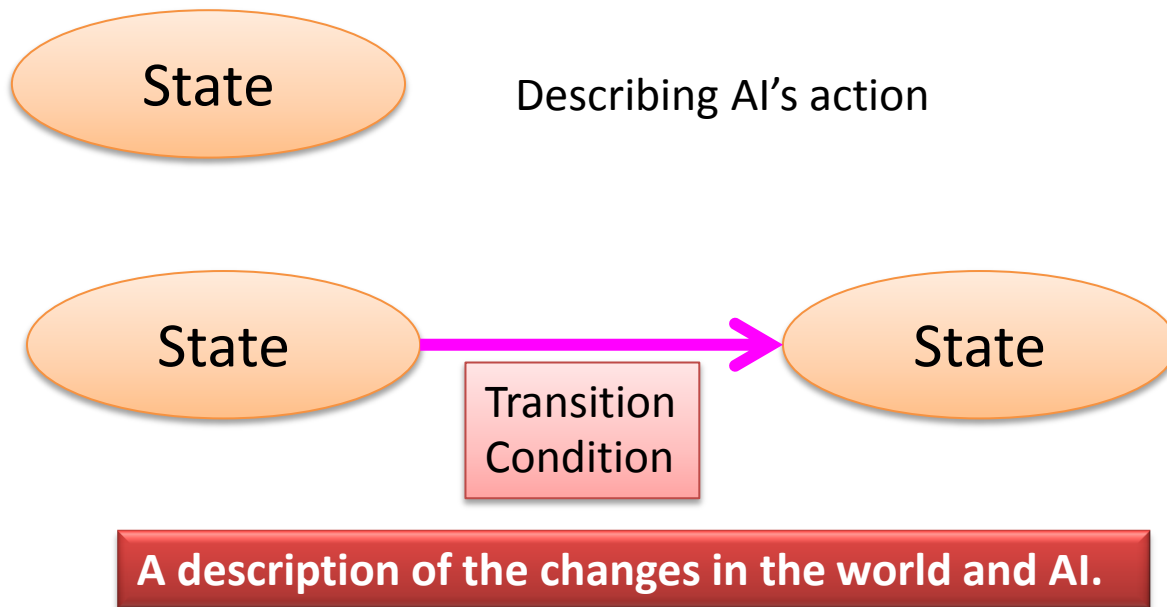
Decision Making is generally a very complex high-degree process. But for digital game there are 7 simple decision making algorithms.



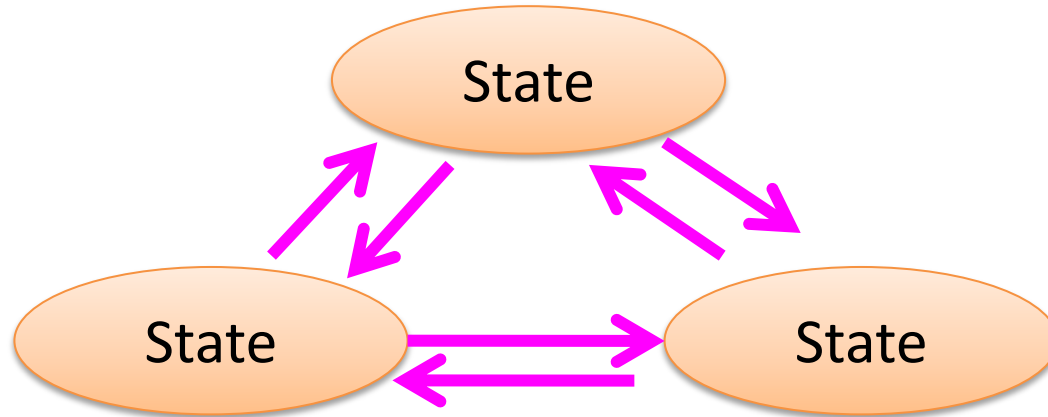
「(something)-based AI」 means that an algorithm uses (something) as a unit.

INTRODUCTION TO STATE MACHINES

State Based AI



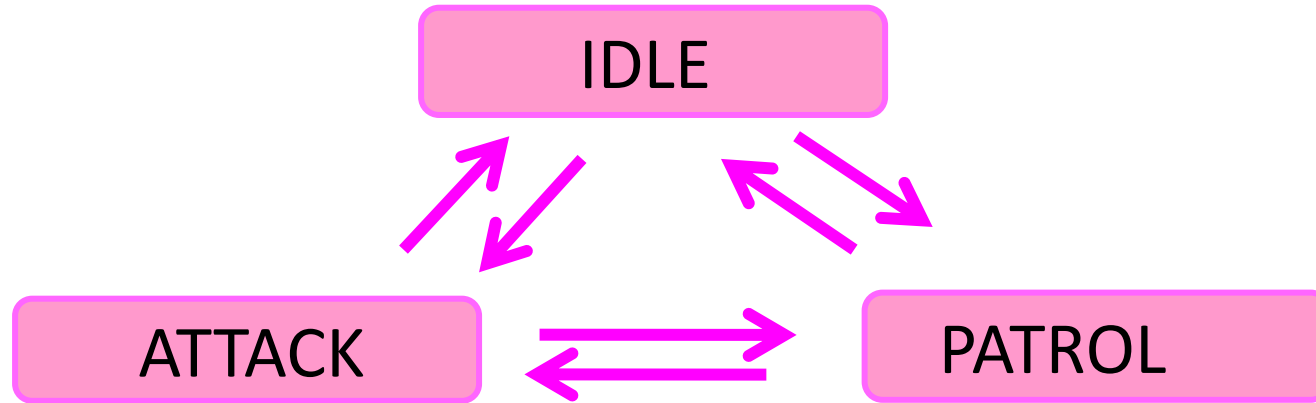
State Machine



State Machine (Finite State Machine)

AI's instruction is described in a state, and changes in the world and AI are described in a transition condition.
A state machine has a loop structure but does not have feedback dynamics.

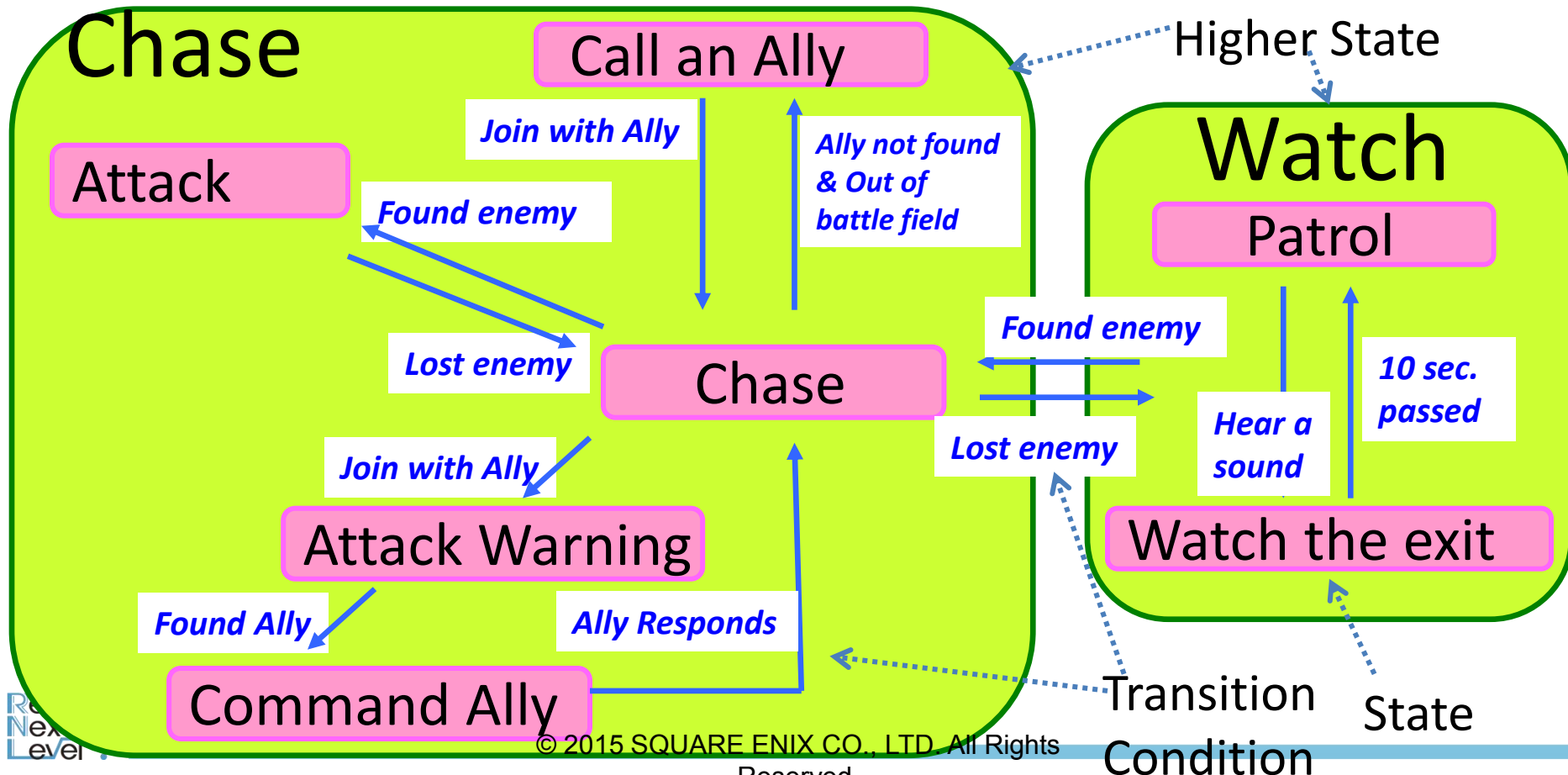
State Machine



State Machine (Finite State Machine)

AI's instruction is described in a state, and changes in the world and AI are described in a transition condition.
A state machine has a loop structure but does not have feedback dynamics.

Hierarchical State Machine



- State machine = Steady control step by step
- Behavior tree = Appropriate adaptive behavior

 I want to use both good points

Agenda



- Introduction to the lecture 3min.
- Introduction to decision-making system (Miyake) 5min.
- **Luminous AI Technical design (Miyake)** 5min.
- AI Graph Editor Implementation (Shirakami) 15min.
- Case study in-game (Namiki, Shirakami) 25min.
- Session summary 2min.

LUMINOUS AI DESIGN POLICY

LUMINOUS AI TECHNICAL DESIGN

- Luminous AI

AI part in Luminous Studio (SQUARE ENIX ORIGINAL ENGINE)

Concept List

- | | | |
|---|--|--|
| <ul style="list-style-type: none"> • Deep participation with the world • Information flow • Body movement flow • Consciousness flow • Structure and flexibility (Learning) • Reaction and delay • Play • Language • Song • Getting affordance • Motion possibility • Simulation -Affordance-Animation • Construction • Collection • Simulation of enemy • Time scale • Space scale • Simulator • Play • Encoding and Image | <ul style="list-style-type: none"> • AI and Animation Simplification、Complexity • Hierarchical nature • Extended body • Description of body • Sensor / Effector • Agent Architecture • Arbiter • Hierarchical Blackboard model • Subsumption • Hierarchical memory | <ul style="list-style-type: none"> • Social • Chemical communication • Social Graph • Social propagation • Communication • Social agent |
| <ul style="list-style-type: none"> • GWT(Global Working Theory) • Dennett's theater model • Stage • Multi stage • Stage switching • Attention | <ul style="list-style-type: none"> • Design pattern • Module • Objective list for each purpose • Object Communication Model | <ul style="list-style-type: none"> • Competition • FSM • Rule-base • Blackboard sharing with enemy and allies • Multiple thinking • Learning fake • Learning • Utility • Task • Goal • Voting system • Scheduling • Blackboard system |
| | <ul style="list-style-type: none"> • Memory • Recall • Confidence • WORKING MEMORY • Active memory • Agent-centric • Situation representation ;/ object representation / knowledge and World representation • Affordance / Action / Effect • Knowledge Engineering | <ul style="list-style-type: none"> • Foreign Structure • Construction AI • Meta Ai |

Concept List

- Deep participation with the world
- Information flow
- Body movement flow
- Consciousness flow
- Structure and flexibility (Learning)
- Reaction and delay
- play
- Language • Song
- Getting affordance
- Motion possibility
- Simulation -Affordance-Animation
- Construction
- Collection
- Simulation of enemy
- Time scale
- Space scale

- Simulator
- Play
- Encoding and Image
- GWT(Global Working Theory)
- Dennett's theater model
- Stage
- Multi stage
- Stage switching
- Attention
- Objective

- AI and Animation Simplification Complex Hierarchical
- Extended body
- Description of body

- Sensor / Effector
- Agent Architecture
- Arbiter
- Hierarchical Blackboard model
- Subsumption
- Hierarchical memory

- Design pattern
- Module
- Objective list for each purpose
- Object Communication Model

- Memory
- Recall
- Confidence
- WORKING MEMORY
- Active memory
- Agent-centric
- Situation representation ;/ object representation / knowledge and World representation
- Affordance / Action / Effect
- Knowledge Engineering
- Memory Fact
- Memory and compression
- Influence map

- Social
- Chemical communication
- Social Graph
- Social propagation
- Communication
- Social agent

- Competition
- FSM
- Rule-base
- Blackboard sharing with enemy and allies
- Multiple thinking
- Learning fake
- Learning
- Utility
- Task
- Goal
- Voting system
- Scheduling
- Blackboard system

- Foreign Structure
- Construction AI

- Meta AI

- Qualitative Physics
- Umwelt
- Affordance
- Urge theory
- Territory

Concept List

- Deep participation into the world
- Information flow
- Body movement flow
- Consciousness flow
- Structure and flexibility (Learning)

Concept

- Getting affordance
- Motion possibility
- Simulation -Affordance-Animation
- Construction
- Collection
- Simulation of enemy
- Time scale
- Space scale

- Simulator

Consciousness Model

- Stage-switching
- Attention
- Objective

Body • Animation

- Extended body
- Description of body

- Sensor / Effector

Architecture

- Hierarchical blackboard model
- Subsumption
- Hierarchical memory

- Design pattern

Design Technology

- Object-Communication model

- Memory
- Recall

Memory

- Situation representation ;/ object representation / knowledge and World representation
- Affordance / Action / Effect
- Knowledge Engineering
- Memory Fact
- Memory and compression
- Influence map

- Social

Social AI

- Communication
- Social agent

- Competition

Decision making algorithm

- Scheduling

Construction AI

- Foreign Structure
- Construction AI

Meta

- Quantitative Physics

Ecological

A
A
A

I
I
I

Concept List

- Deep participation into the world
- Information flow
- Body movement flow
- Consciousness flow
- Structure and flexibility (Learning)

Concept

Getting affordance

Body • Animation

- Extended body
- Description of body

Sensor / Effector

Architecture

Hierarchical blackboard model

Substrate

- Social

Social AI

- Communication
- Social agent

Competition

Decision

Luminous AI will include all main concepts in order.

Simulator

Consciousness Model

Stage Switching

Attention

Objectives

Memory

- Situation representation ;/ object representation / knowledge and World representation
- Affordance / Action / Effect
- Knowledge Engineering
- Memory Fact
- Memory and compression
- Influence map

Construction AI

Foreign Structure

Construction AI

Meta

Quantitative Physics

Ecological

A

I

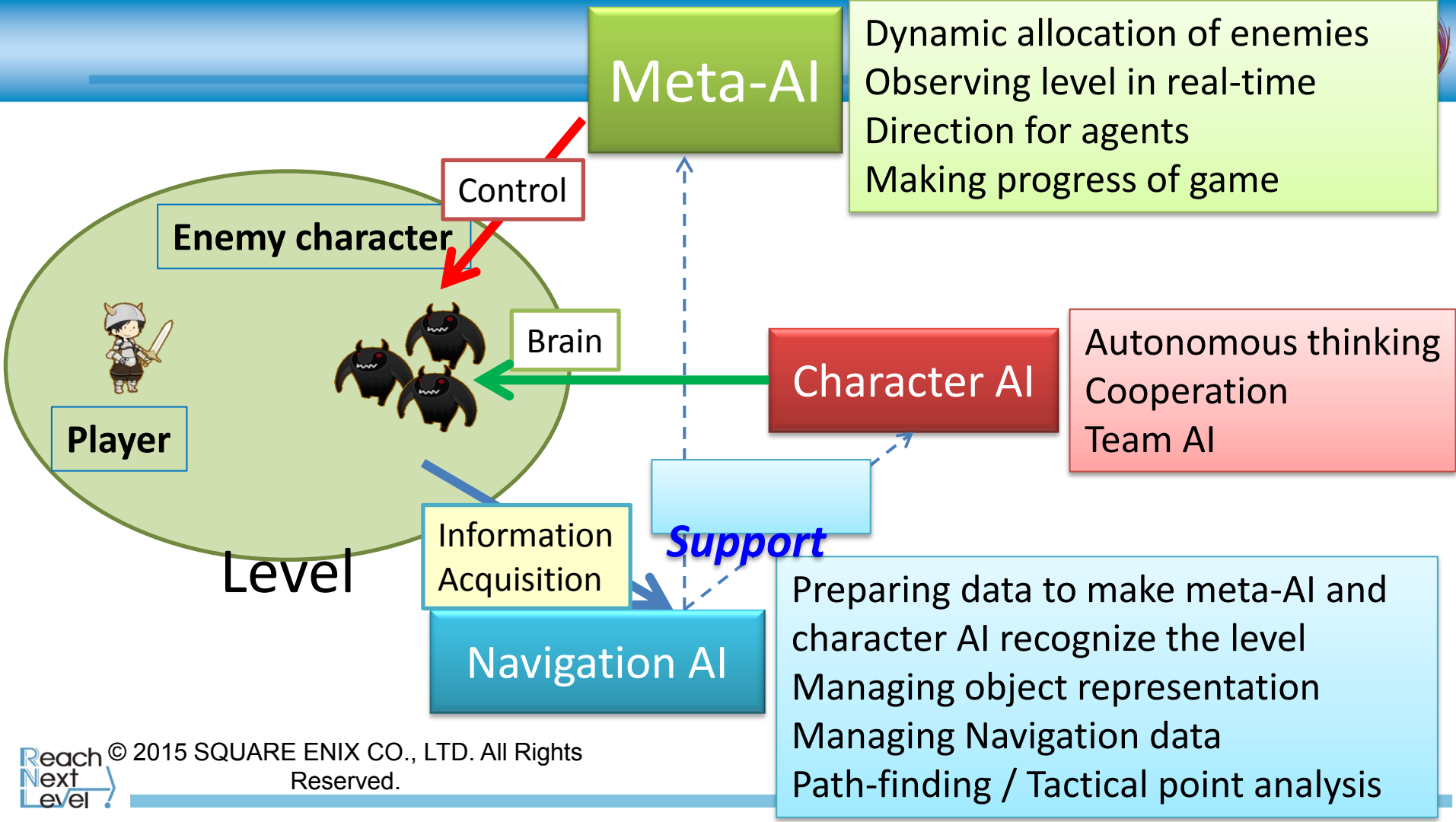
A

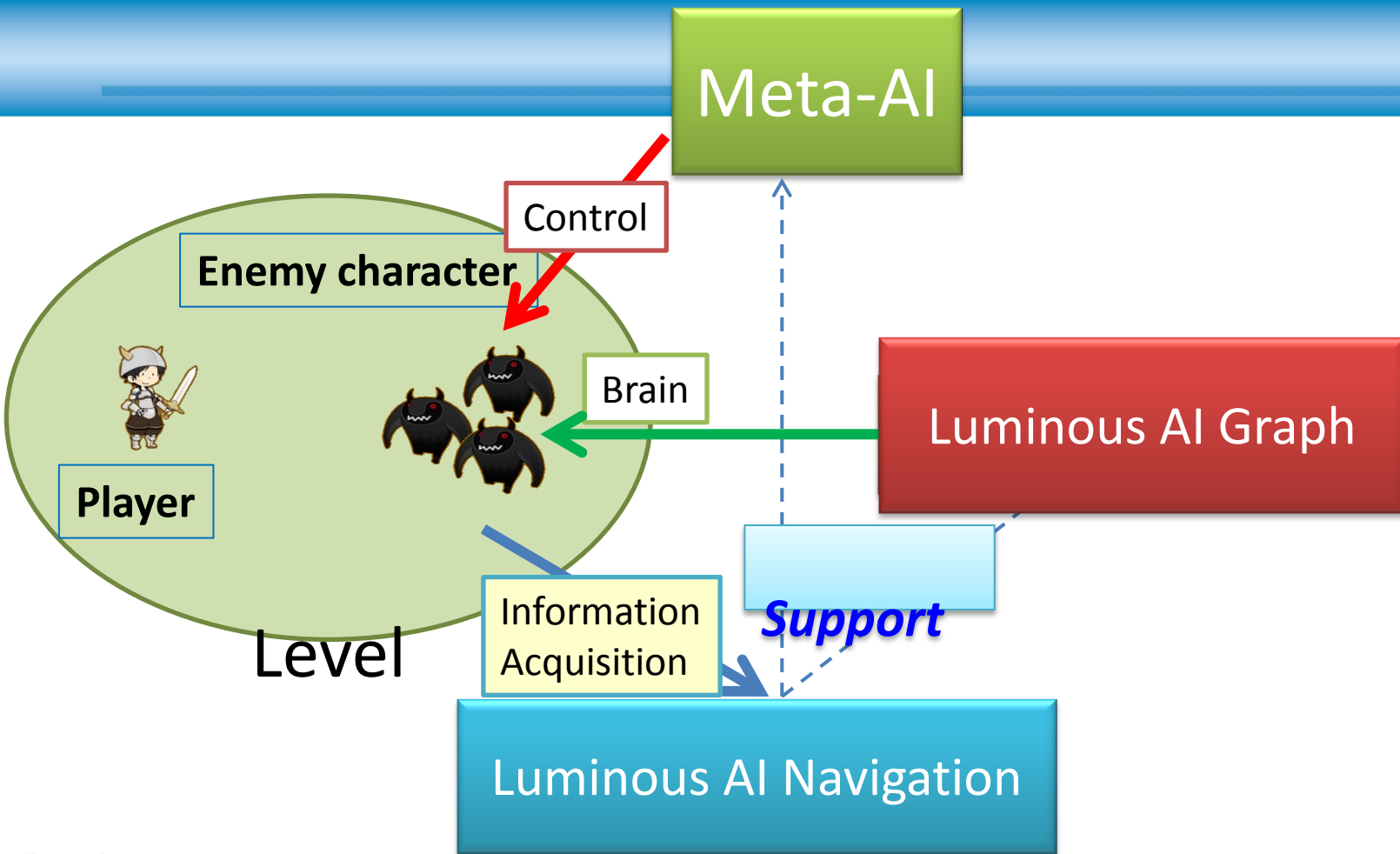
T

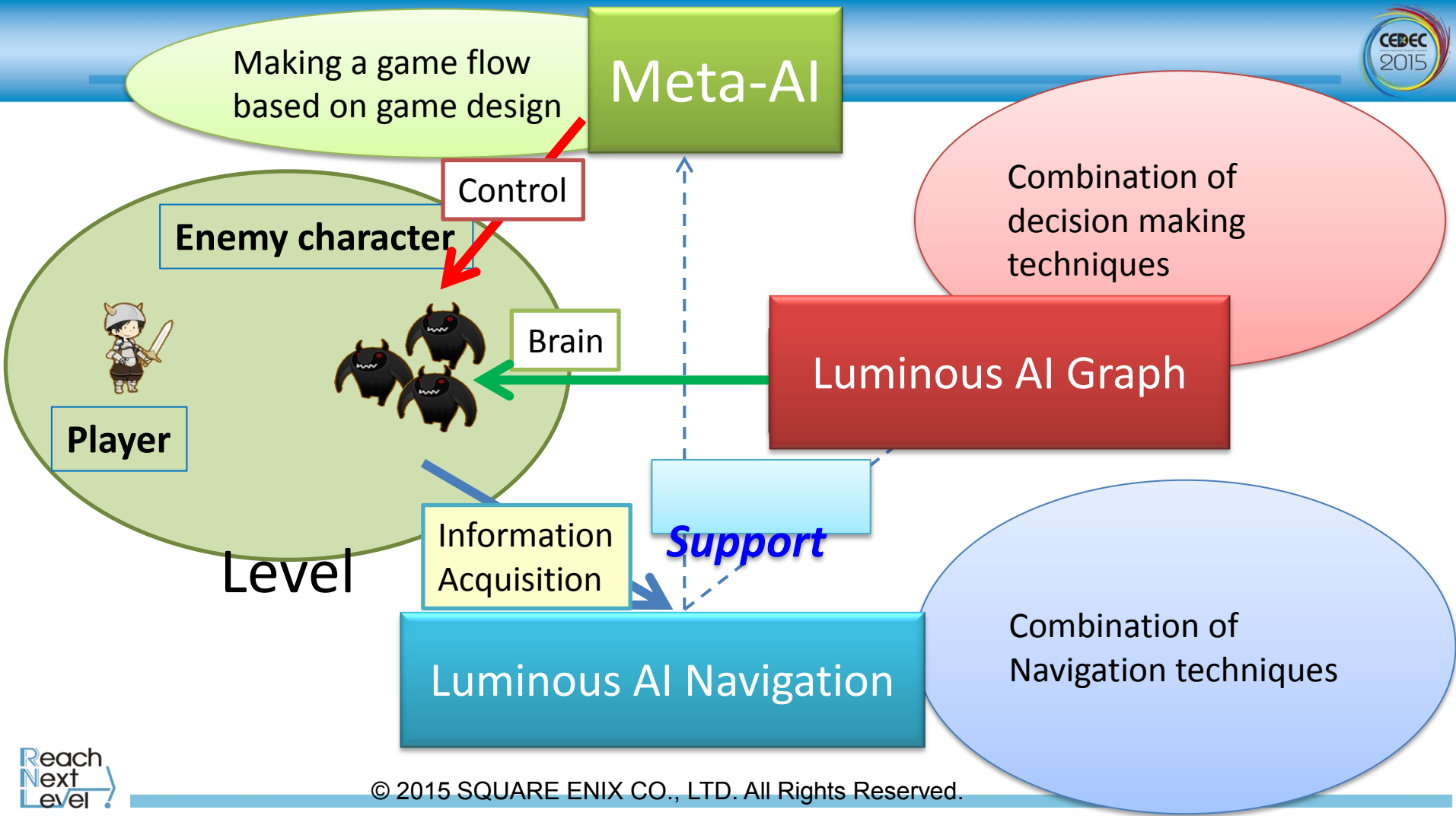
Game Engine

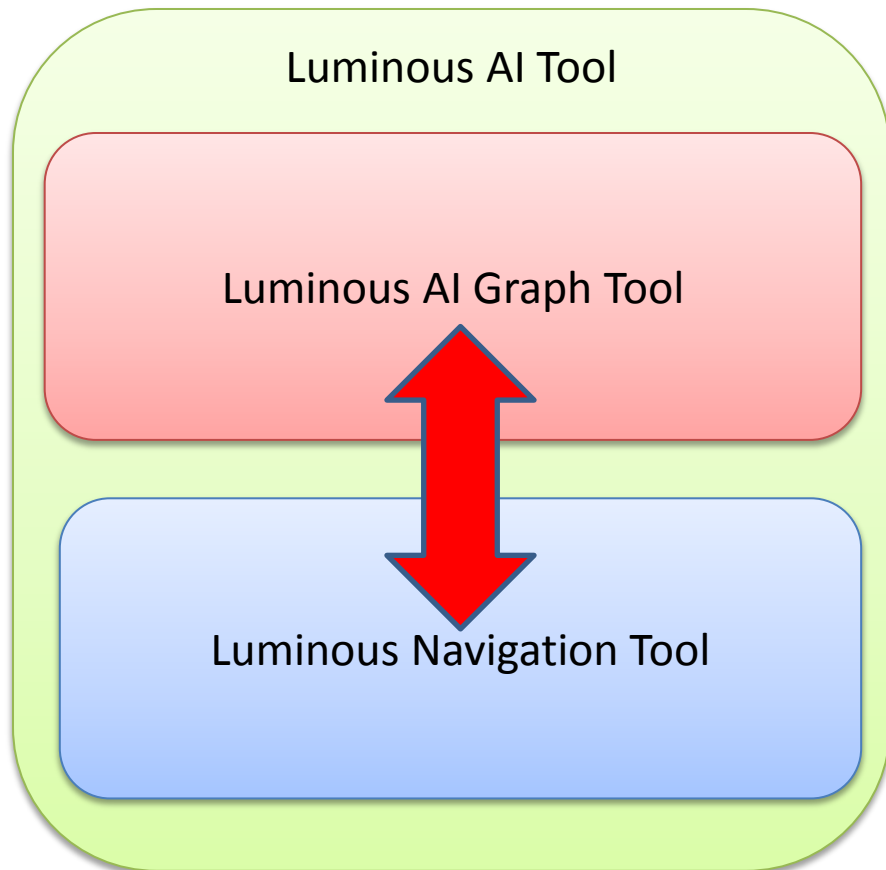


- Game Engine
= where the company's technologies are gathered
- Luminous AI
= where the company's AI technologies are gathered

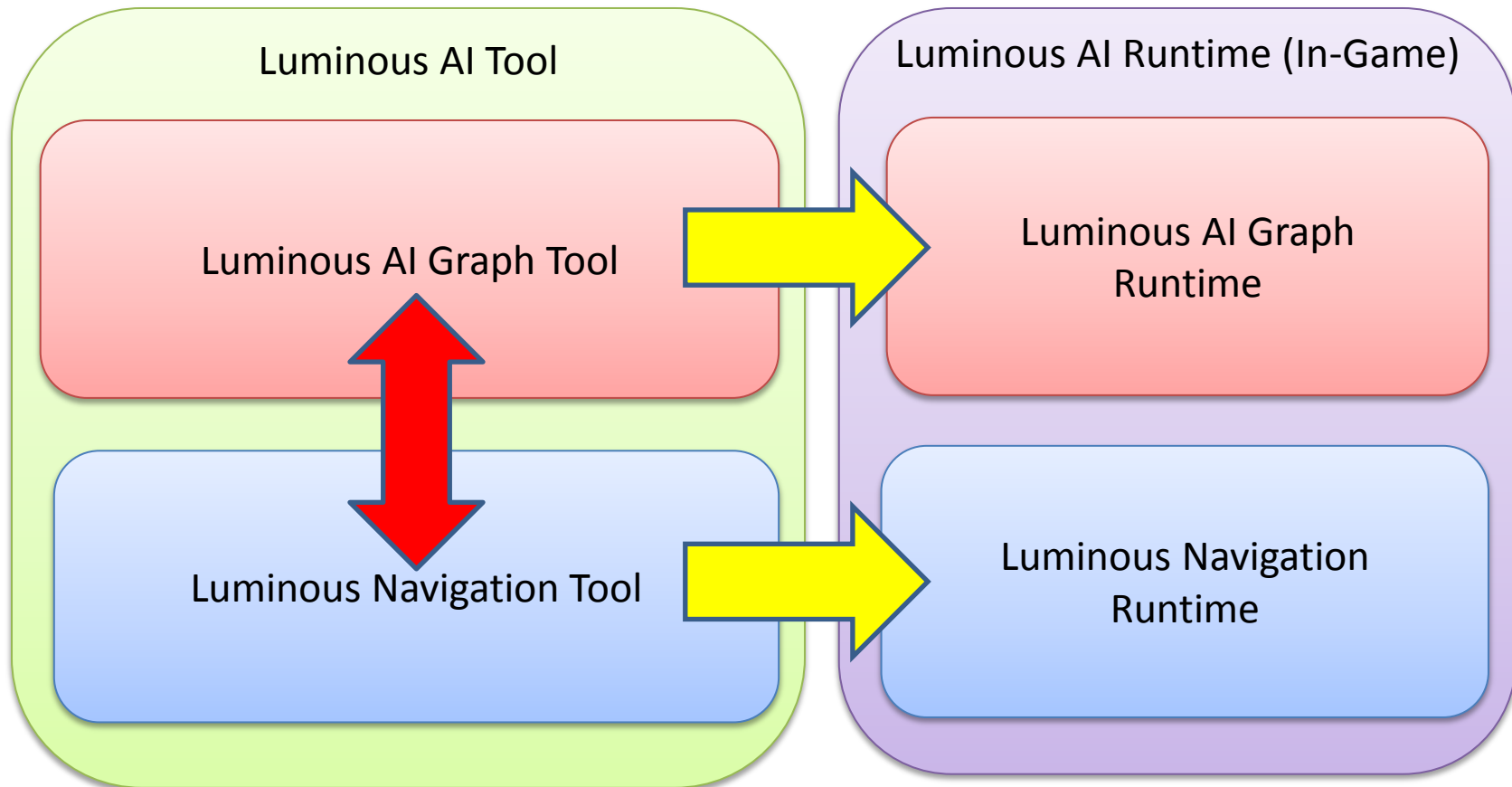




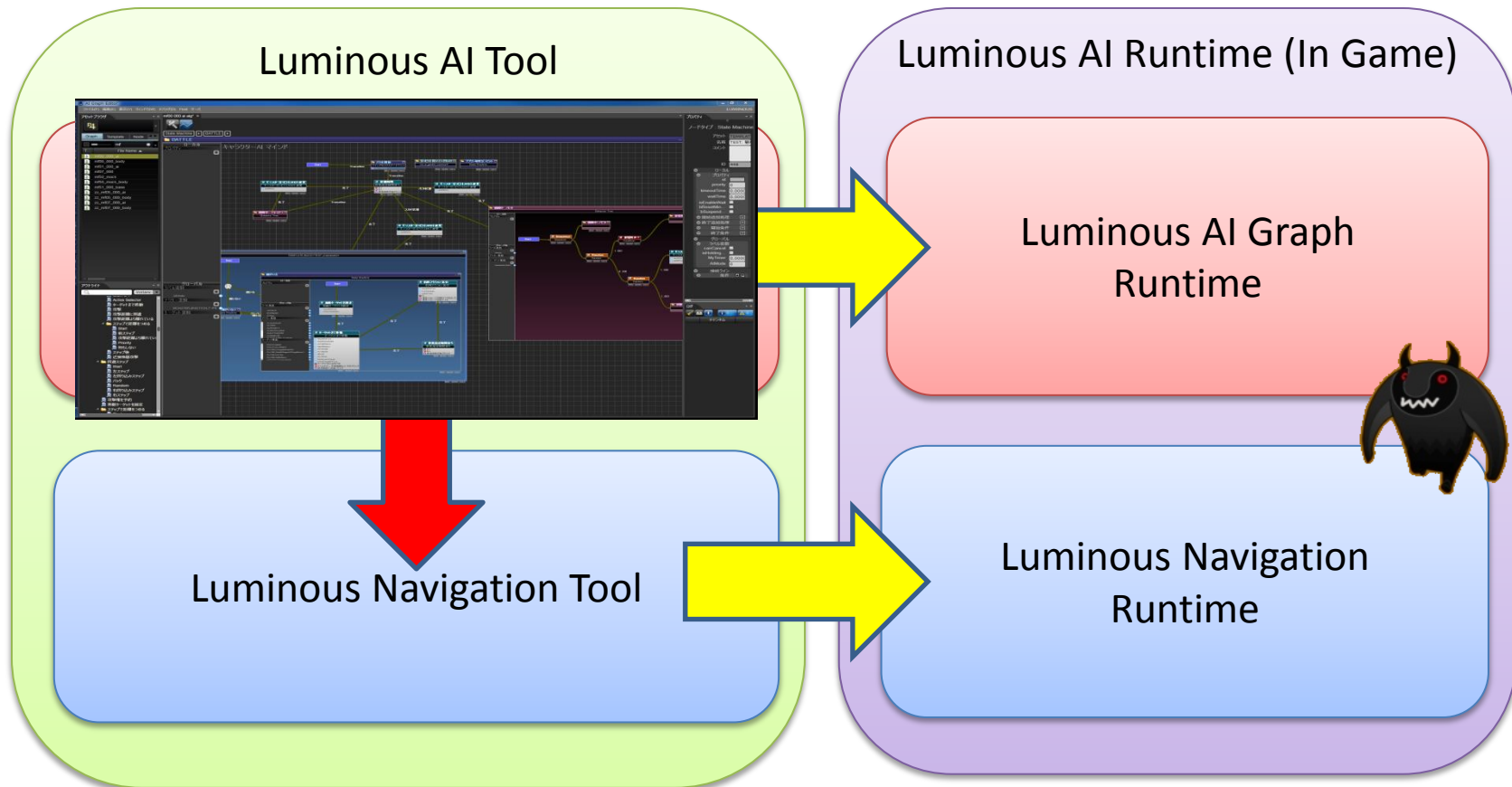




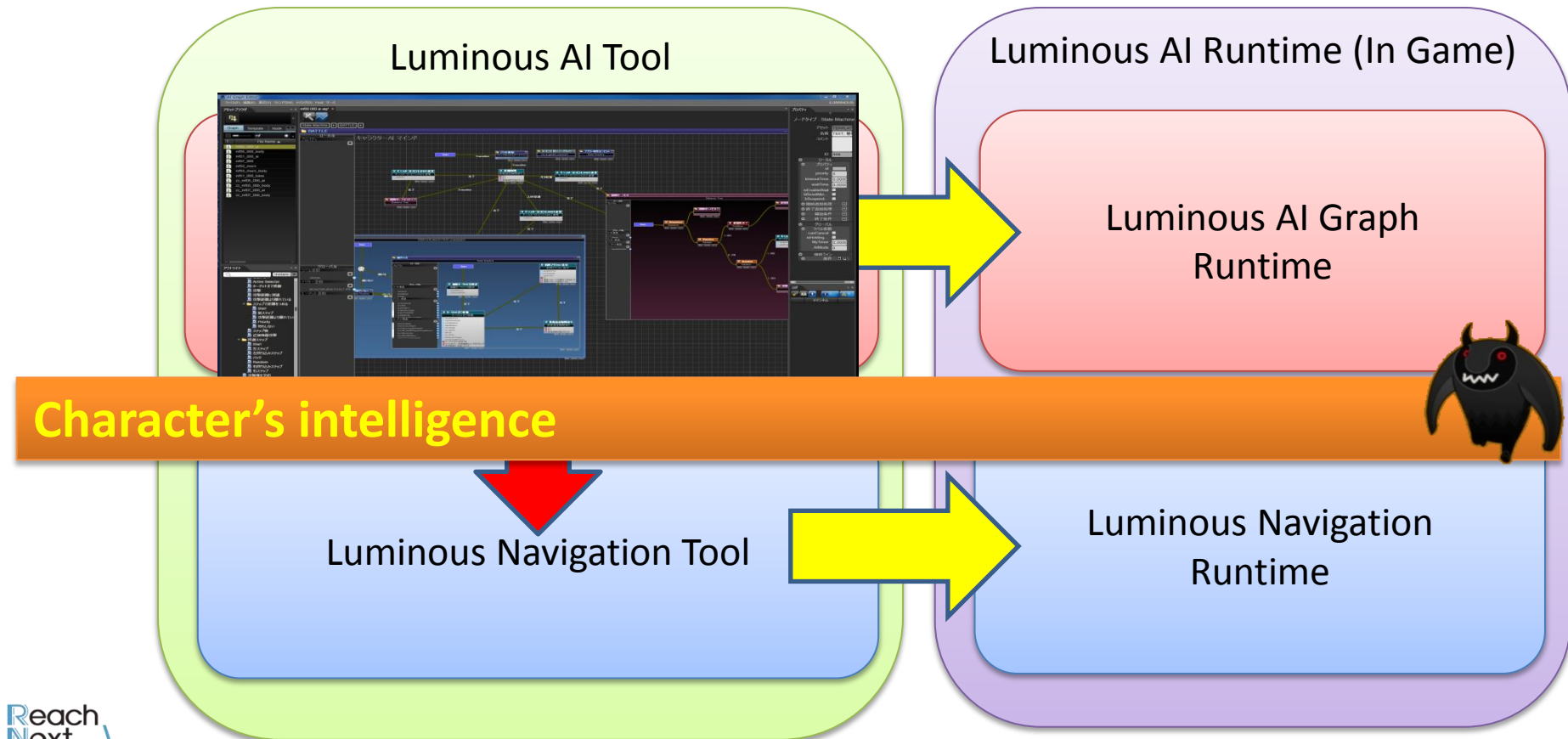
Luminous AI Architecture



Luminous AI Architecture

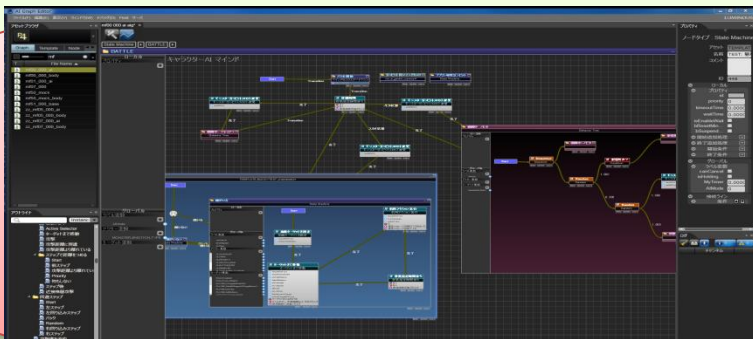


Luminous AI Architecture



Luminous AI Architecture

Luminous AI Tool



Luminous AI Runtime (In Game)

Luminous AI Graph Runtime

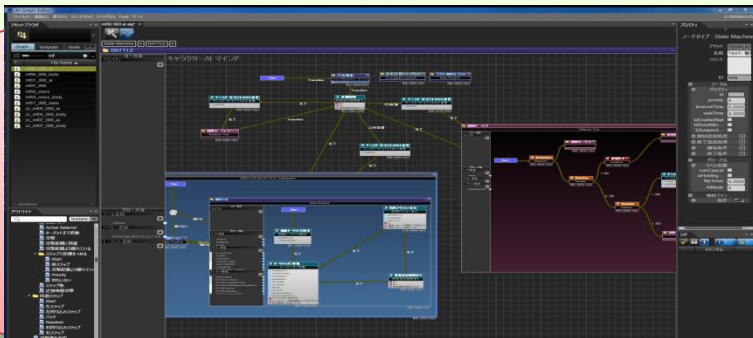


Character's intelligence

can be created by **planners or designers** using a graph-based GUI tool.

Luminous AI Architecture

Luminous AI Tool



Luminous AI Runtime (In Game)

Luminous AI Graph Runtime



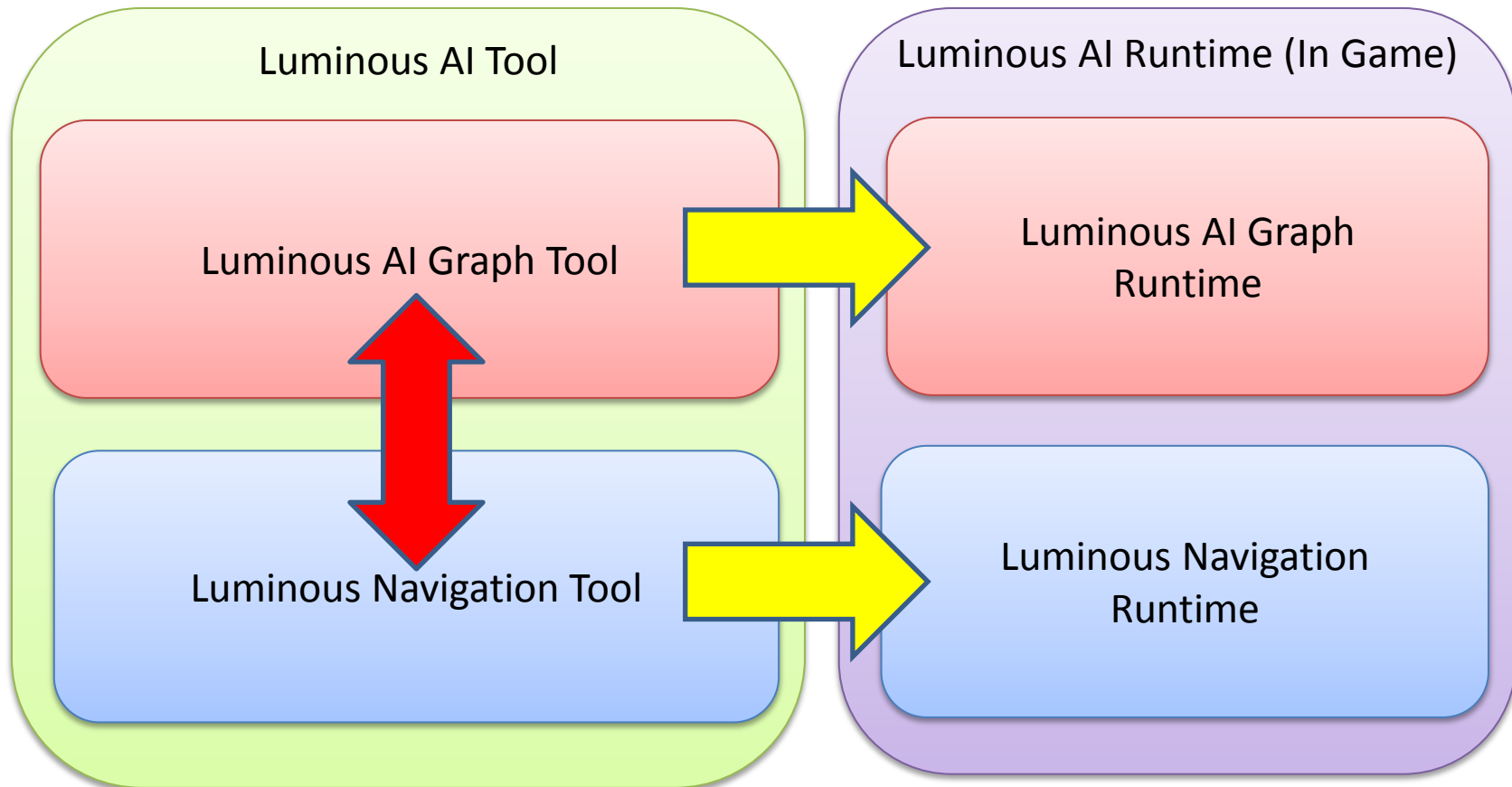
Character's intelligence

can be created by **planners or designers** using a graph-based GUI tool.

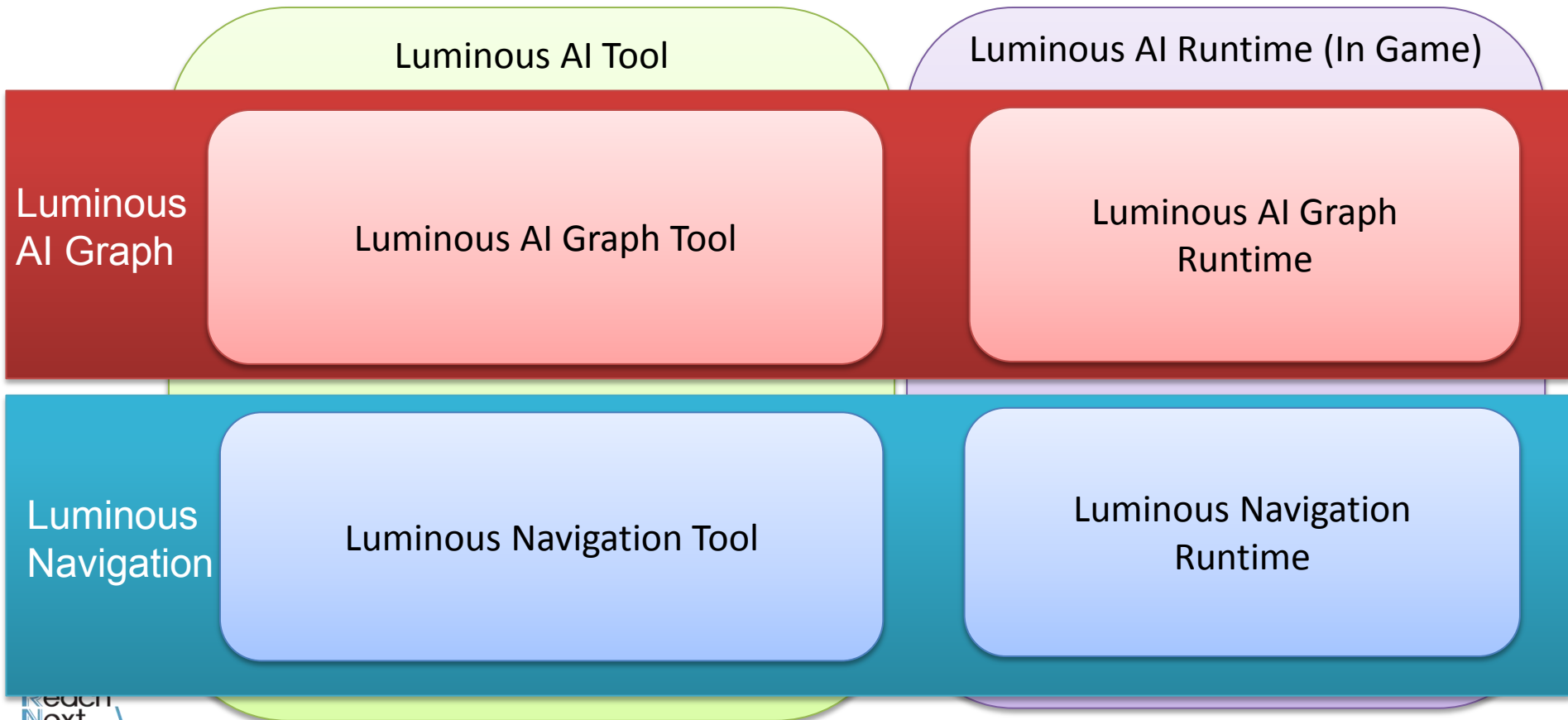


GUI tools make it possible to **separate non-engineering work** from **engineering work**.

Luminous AI Architecture



Luminous AI Architecture



Luminous AI Architecture

Luminous AI Tool

Luminous AI Runtime (In Game)

Luminous AI Graph

Rule-based

Goal-based

Memory

Sensor

State-based

Task-based

Knowledge Representation

Behavior-based

Simulation-based

Blackboard

Utility-based

Target Selection

Learning • Evolution

Luminous Navigation

Heat-map System

Knowledge / World Representation

PQS (Point Query System)

Cover-point System

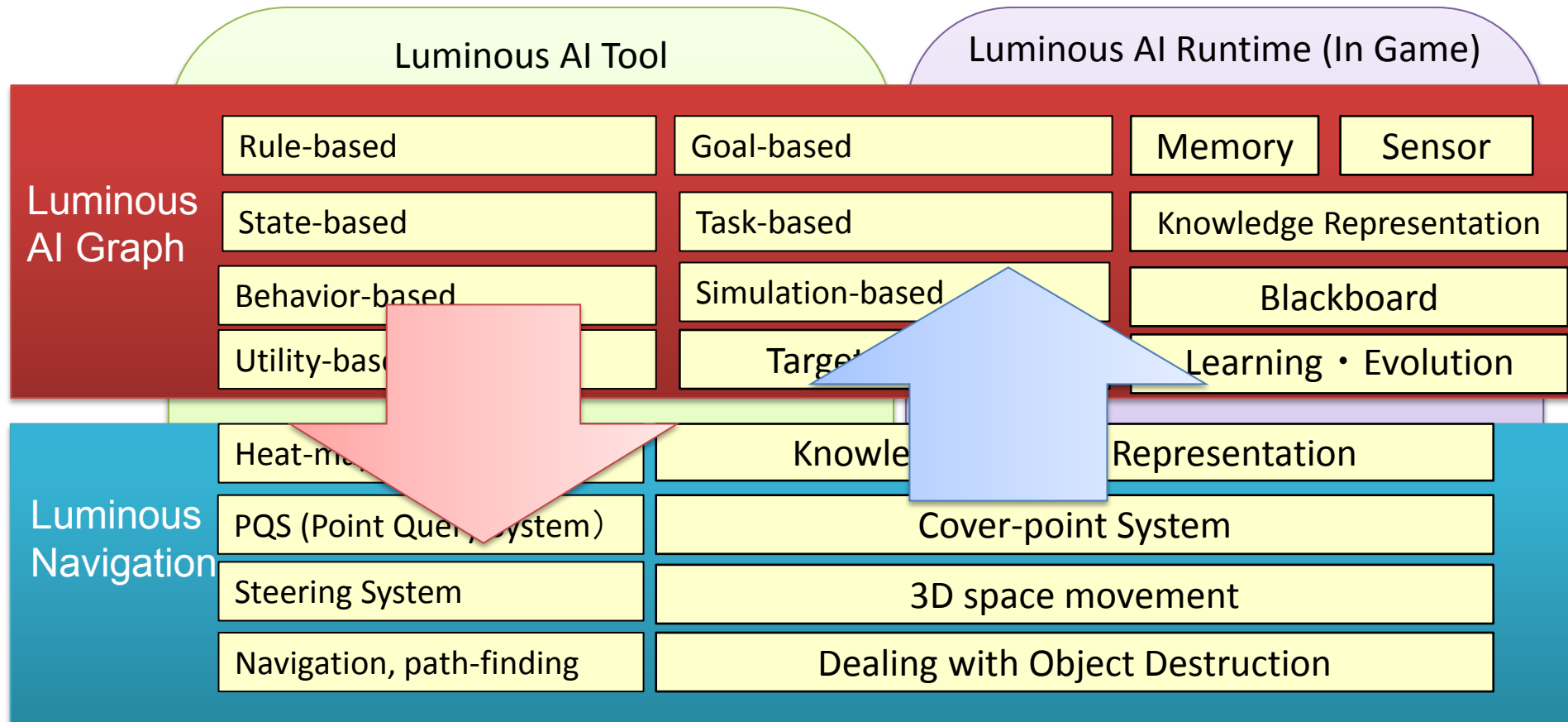
Steering System

3D space movement

Navigation, path-finding

Dealing with Object Destruction

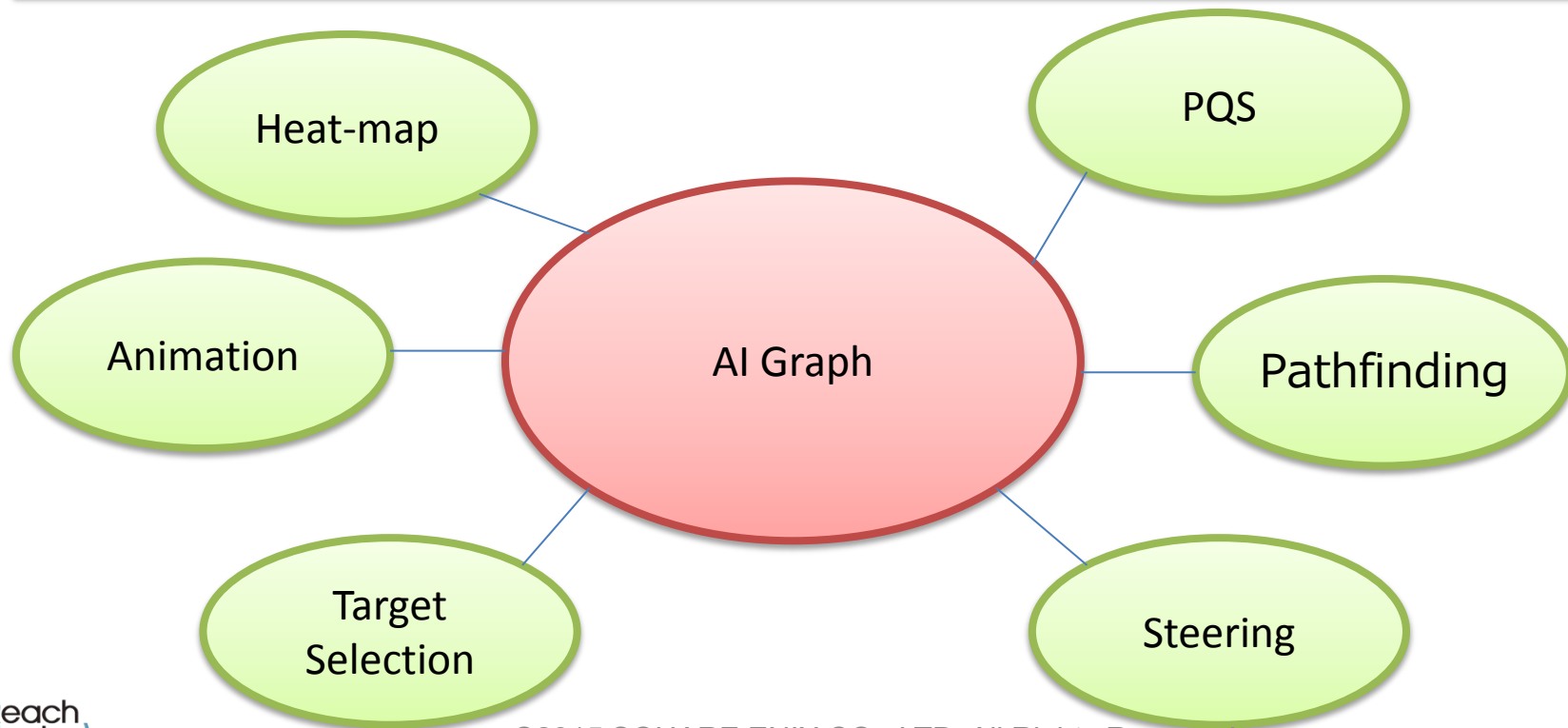
Luminous AI Architecture



Luminous AI Graph connects AI tech



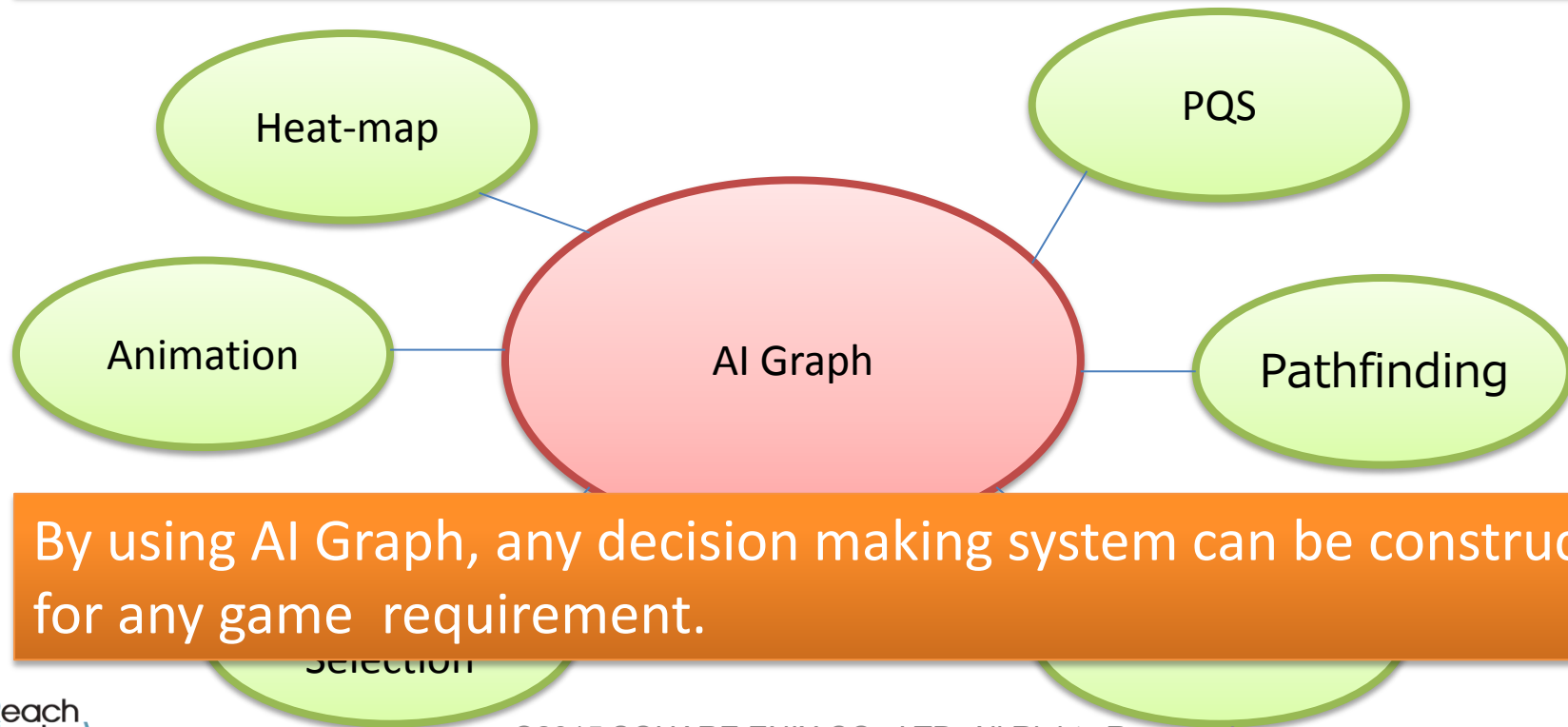
We designed AI Graph to call and connect all functional modules.



Luminous AI Graph connects AI tech



We designed AI Graph to call and connect all functional modules.



By using AI Graph, any decision making system can be constructed for any game requirement.

One decision making algorithm is not enough.



Combining multiple decision making algorithms

State Machine and Behavior Tree

Luminous AI Tool

Luminous AI Runtime (In Game)

Luminous AI Graph

Rule-based

Goal-based

Memory

Sensor

State-based

Task-based

Knowledge Representation

Behavior-based

Simulation-based

Blackboard

Utility-based

Target Selection

Learning • Evolution

Luminous Navigation

Heat-map System

Knowledge / World Representation

PQS (Point Query System)

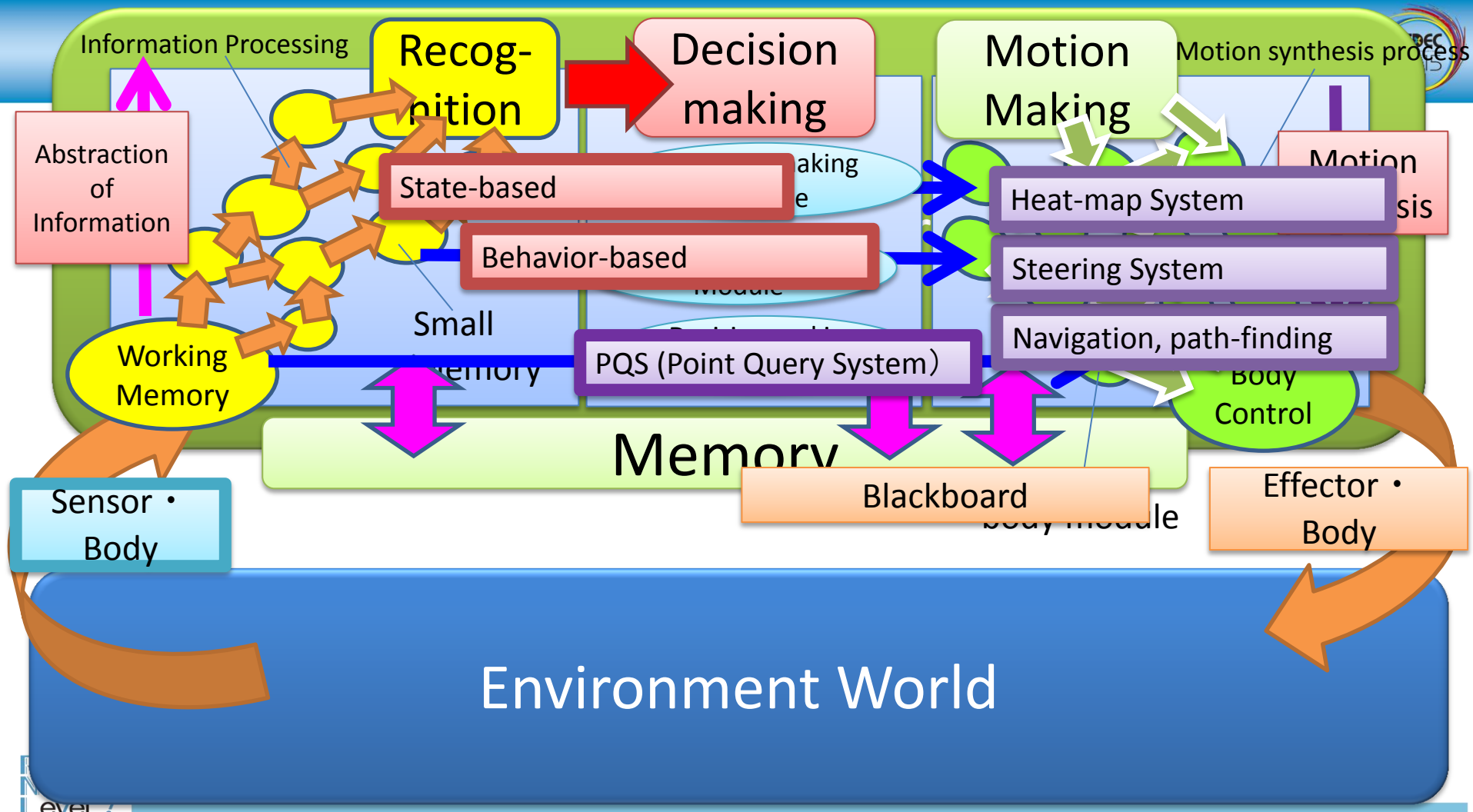
Cover-point system

Steering System

3D space movement

Navigation, path-finding

Dealing with Object Destruction



The Future of Luminous AI

Luminous AI Tool

Luminous AI Runtime (In Game)

Luminous AI Graph

Rule-based

Goal-based

Memory

Sensor

State-based

Task-based

Knowledge Representation

Behavior-based

Simulation-based

Blackboard

Utility-based

Target Selection

Learning • Evolution

Luminous Navigation

Heat-map System

Knowledge / World Representation

PQS (Point Query System)

Cover-point system

Steering System

3D space movement

Navigation, path-finding

Dealing with Object Destruction

Agenda



- Introduction to the lecture 3min.
- Introduction to decision-making system (Miyake) 5min.
- Luminous AI Technical design (Miyake) 5min.
- **AI Graph Editor Implementation (Shirakami) 15min.**
- Case study in-game (Namiki, Shirakami) 25min.
- Session summary 2min.

AI Graph Editor Design and Implementation

AI Graph Editor Design and Implementation



- 4 problems for tool design
- Some functions required from game

AI Graph Editor Design and Implementation



- 4 problems for tool design
- Some functions required from game

4 problems for tool design



1. How to combine behavior tree and state machine ?
2. How to design data structure to have extensibility ?
3. How to make smooth communication between game system and AI system ?
4. How to realize mass production of AI data without copy & paste ?

Problem 1

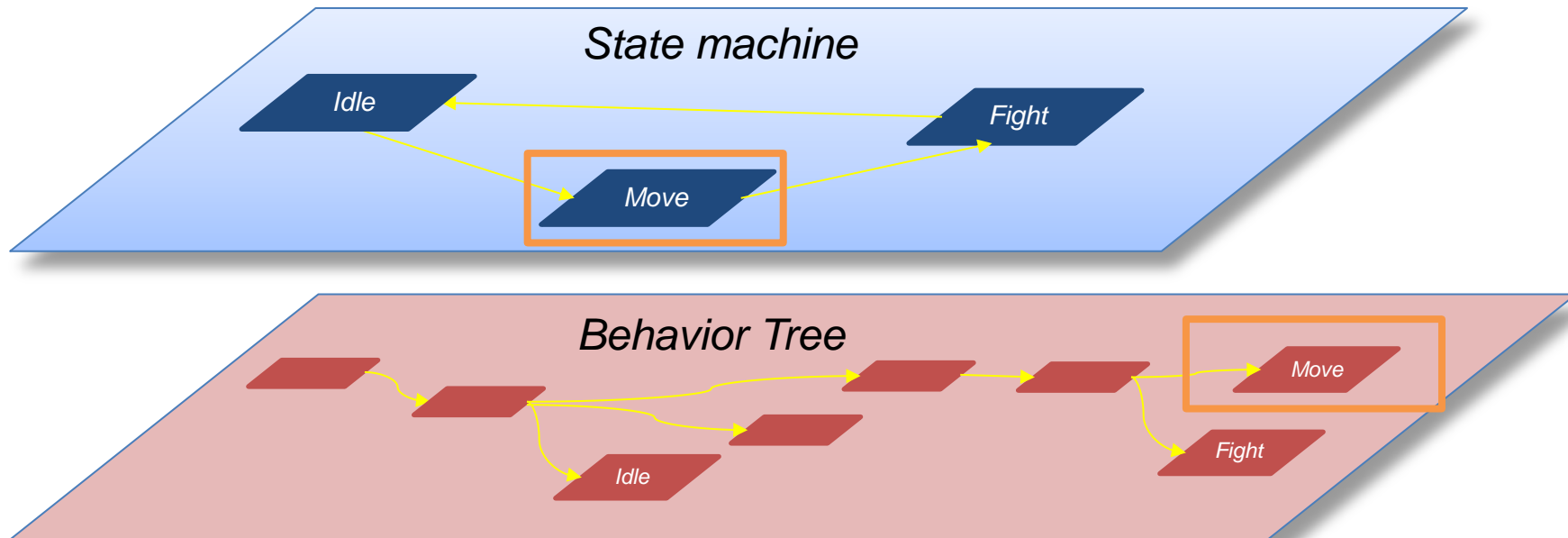


1. How to combine behavior tree and state machine ?

We set two kinds of node: one kind of node is for graph structure, and the other node is for processing.

⇒ **Hybrid Node format**

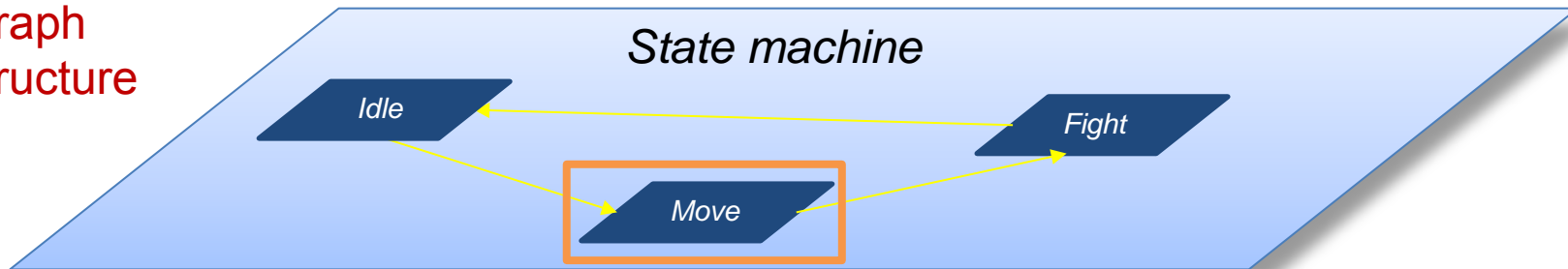
Hybrid node format



- We want to use a combination of state machine and behavior tree which have different graph structures
- Processing in a node is common to both graphs.

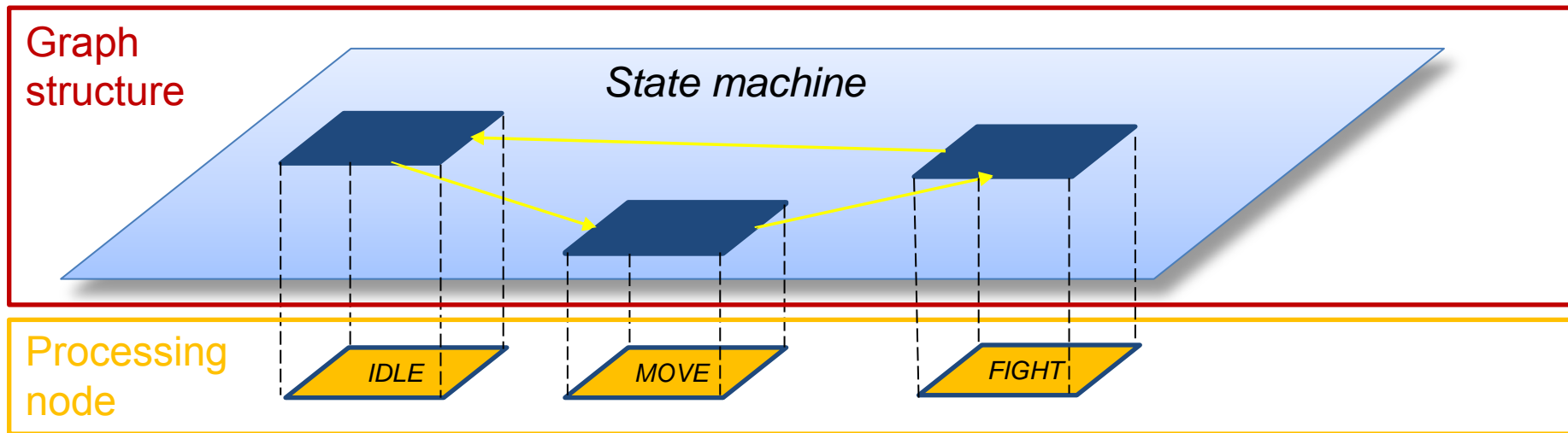
Hybrid node format

Graph
structure



Graph structure

Hybrid node format

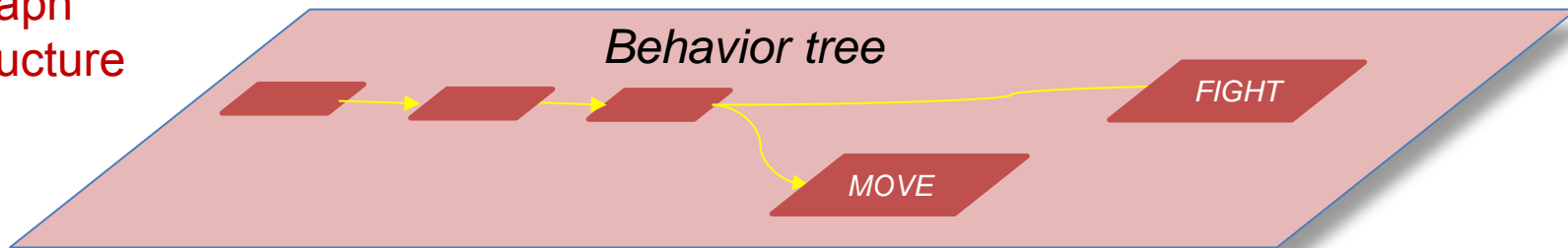


Graph structure is separate from **Processing node**.

A node on **Graph structure** is indirectly related to a node for **Processing**.

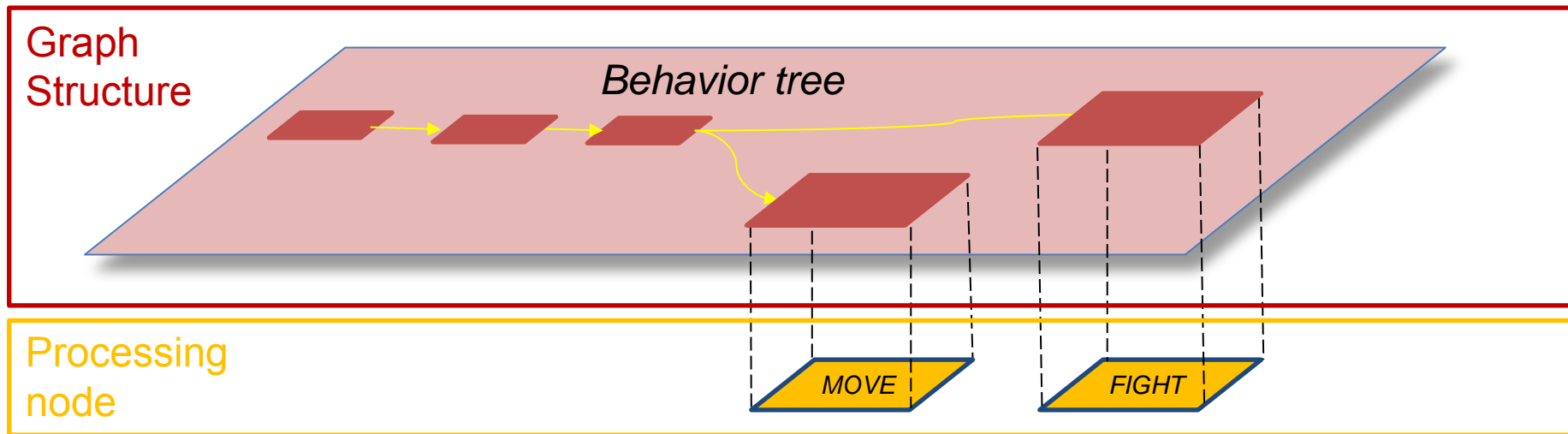
Hybrid node format

Graph
structure



Graph structure

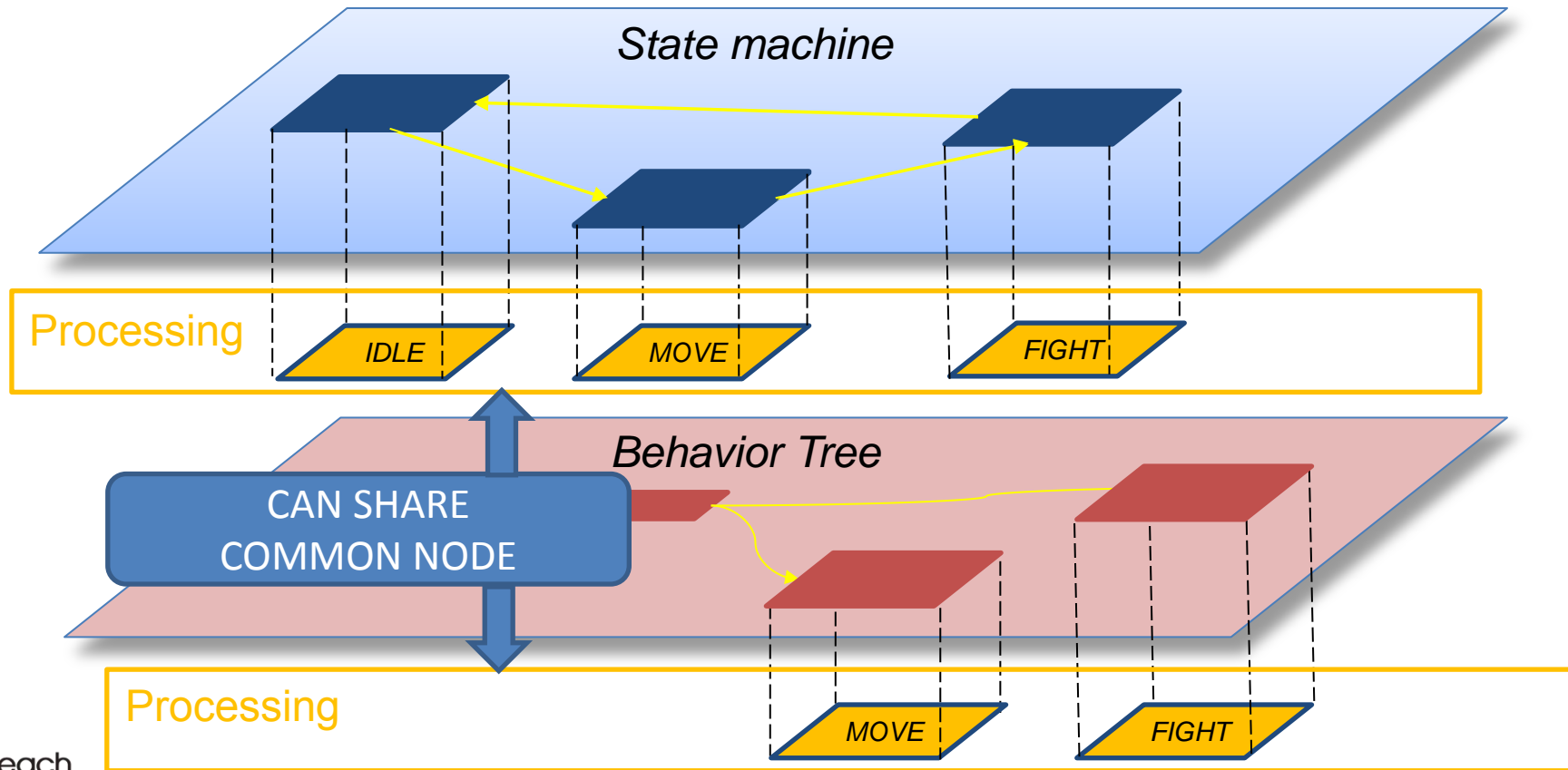
Hybrid node format



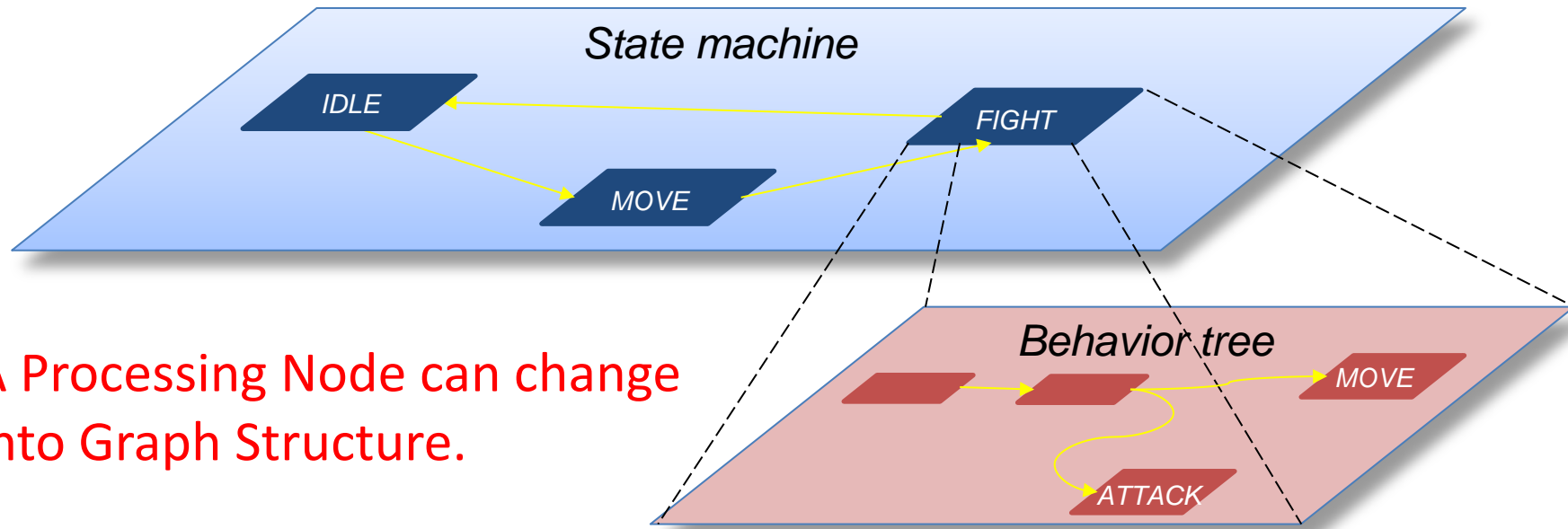
Graph structure is separate from **Processing node**.

A node on **Graph structure** is indirectly related to a node for **Processing**.

Hybrid node format



Hybrid node format

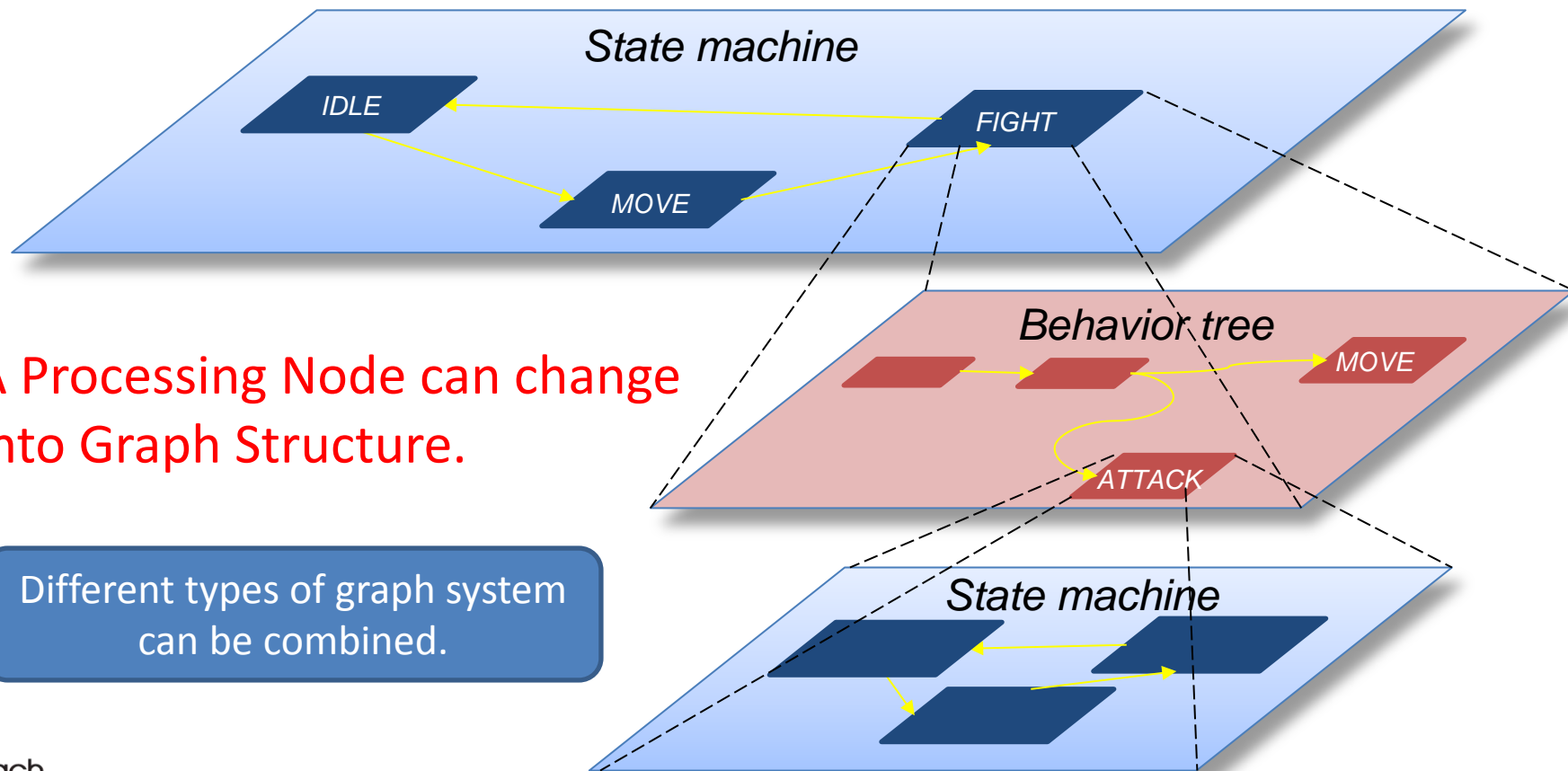


A Processing Node can change
into Graph Structure.

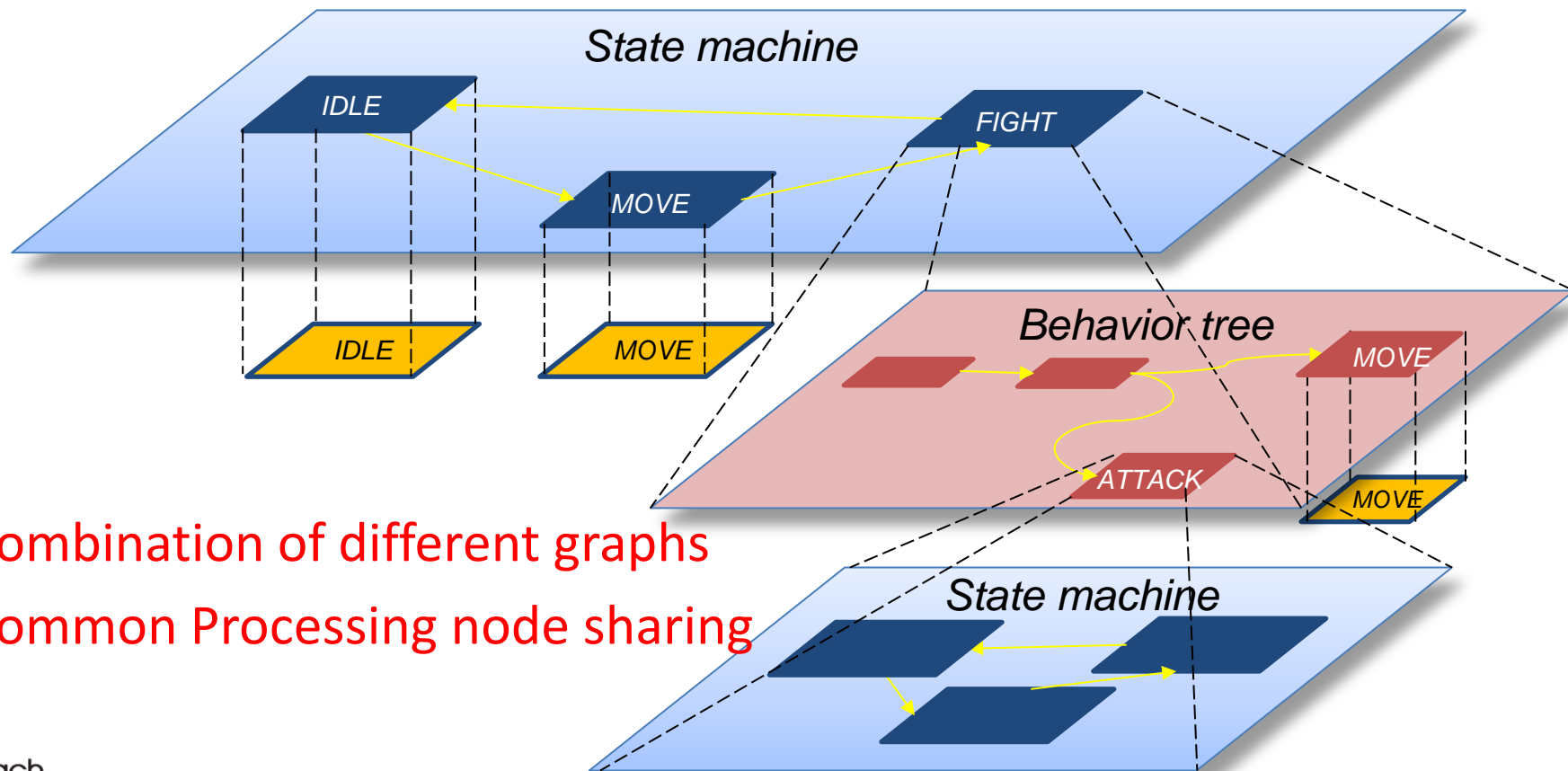
Hybrid node format

A Processing Node can change
into Graph Structure.

Different types of graph system
can be combined.

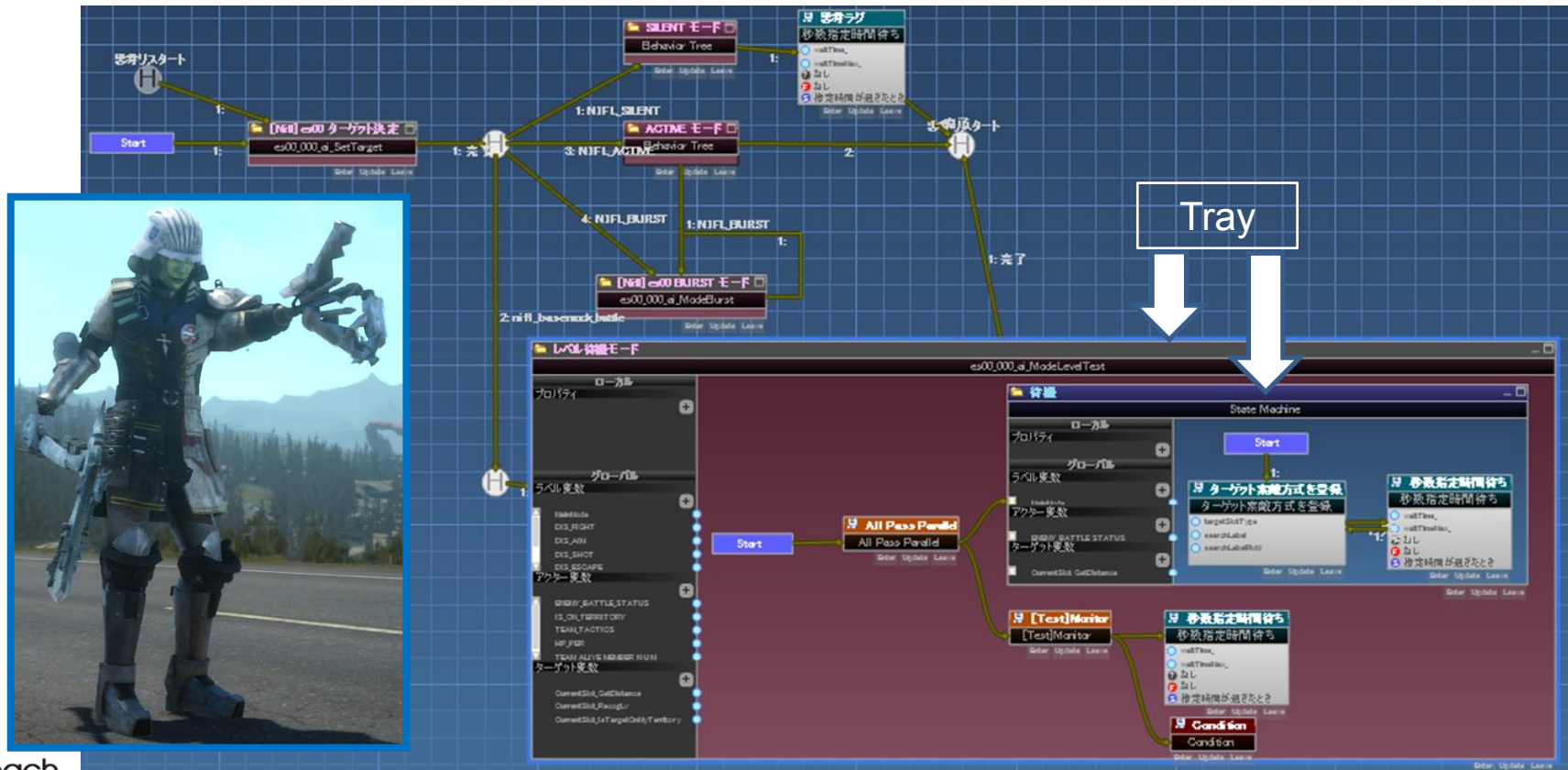


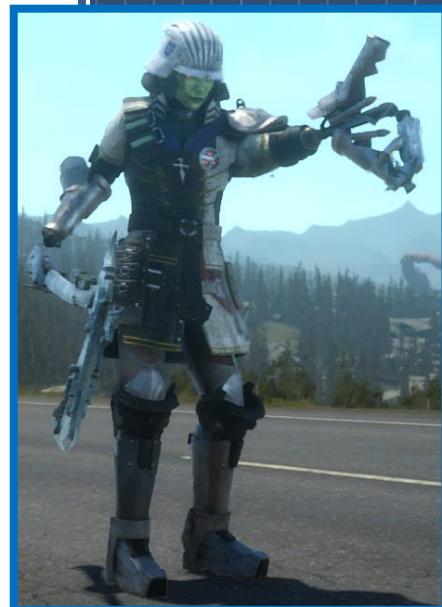
Hybrid node format



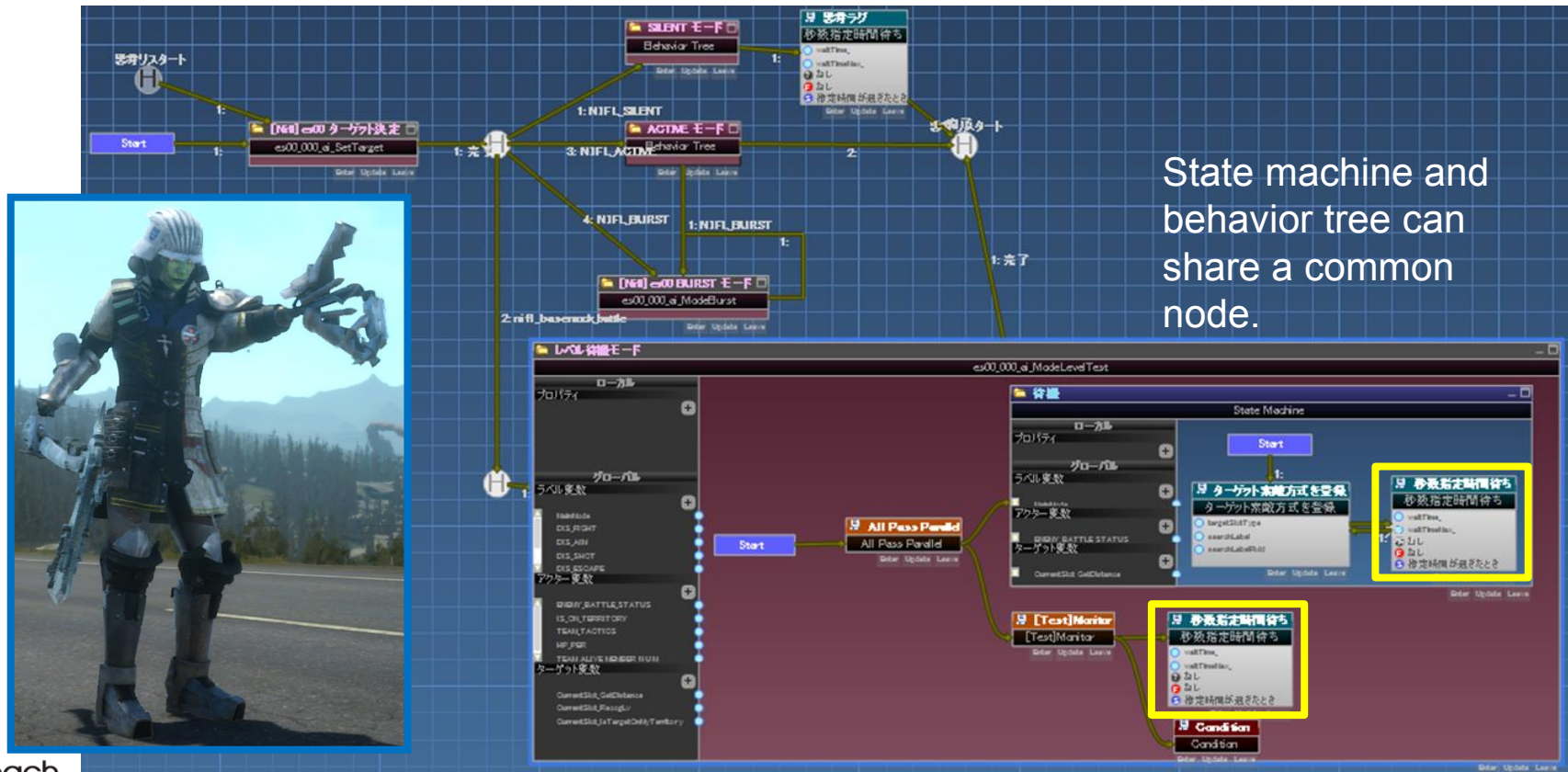
- Combination of different graphs
- Common Processing node sharing

Hybrid node format





Hybrid node format



Problem 2



2. How to design data structure to have extensibility ?

Hierarchical graph system makes hierarchical data structure possible.

⇒ Hierarchical scalability

Scalability by Hierarchy



- Enlarging data size should be avoided (caused by combination of nodes).
- We want to quickly accommodate changes to the game's requirements.

Scalability by Hierarchy



- Enlarging data size should be avoided (caused by combination of nodes).
- We want to quickly accommodate changes to the game's requirements.

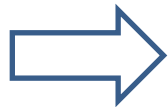


Build a **Hierarchy** from tray units
+ divide data by **appropriate data size**

Scalability by Hierarchy



- Enlarging data size should be avoided (caused by combination of nodes).
- We want to quickly accommodate changes to the game's requirements.



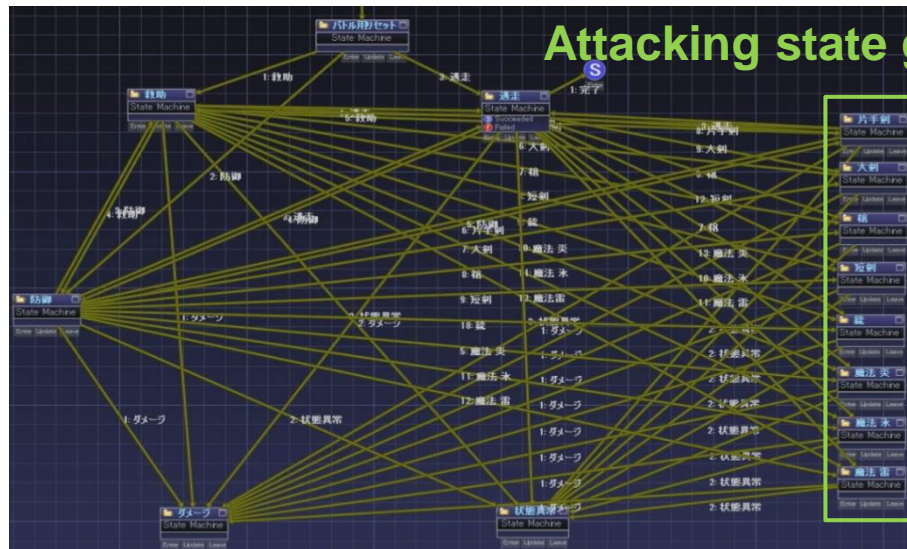
Build a **Hierarchy** from tray units
+ divide data by **appropriate data size**

It gives high scalability by reducing combinational explosion and easily adding new nodes.

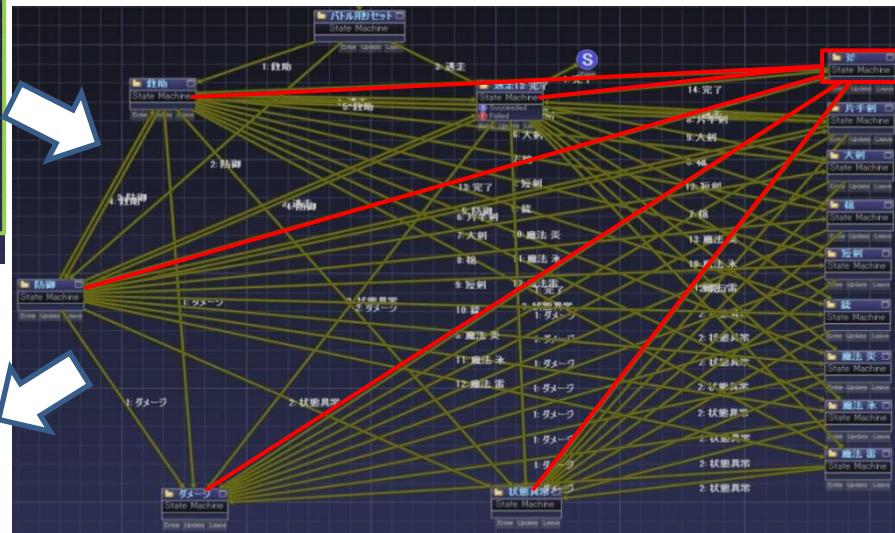
Scalability by Hierarchy

Attacking state group

When one attacking state is added, transition to state must be added.

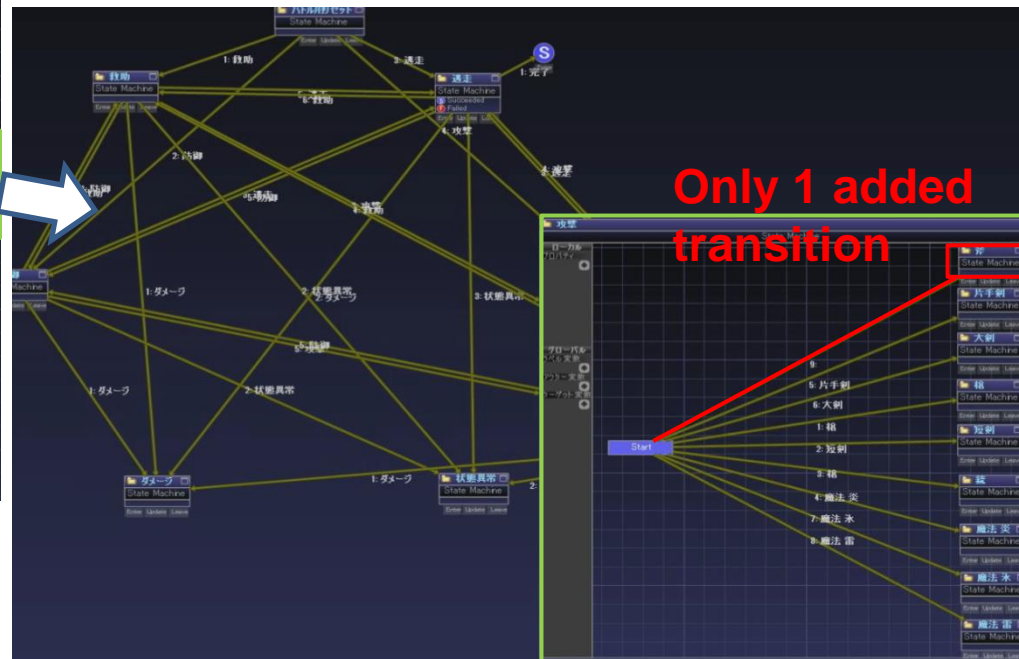
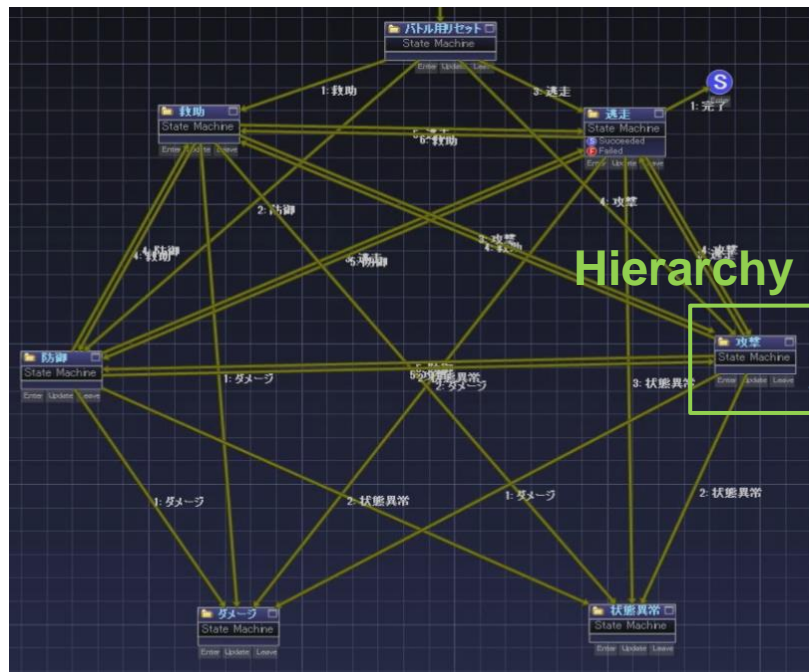


When number of states increase, number of transitions increase...

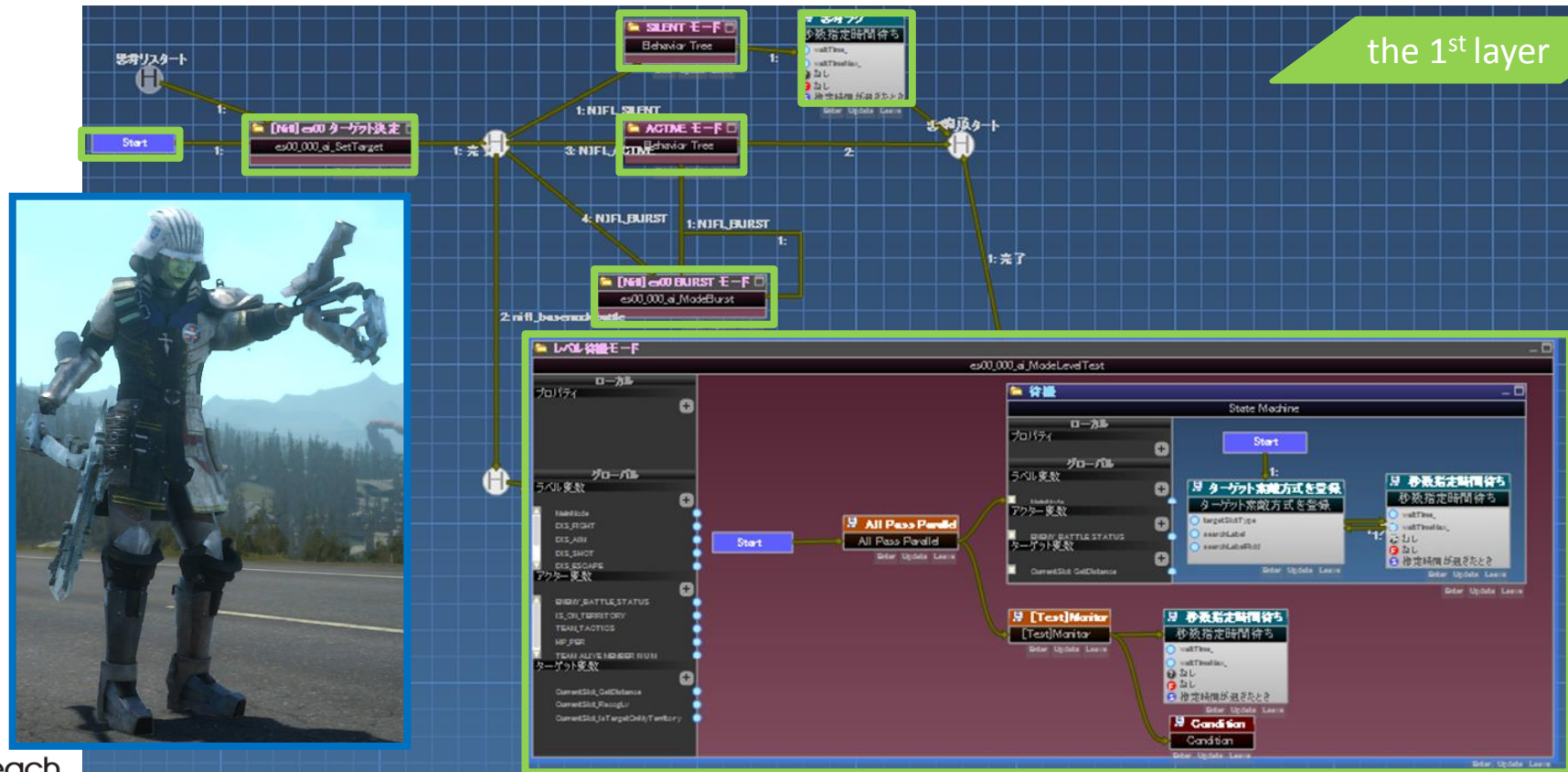


Scalability by Hierarchy

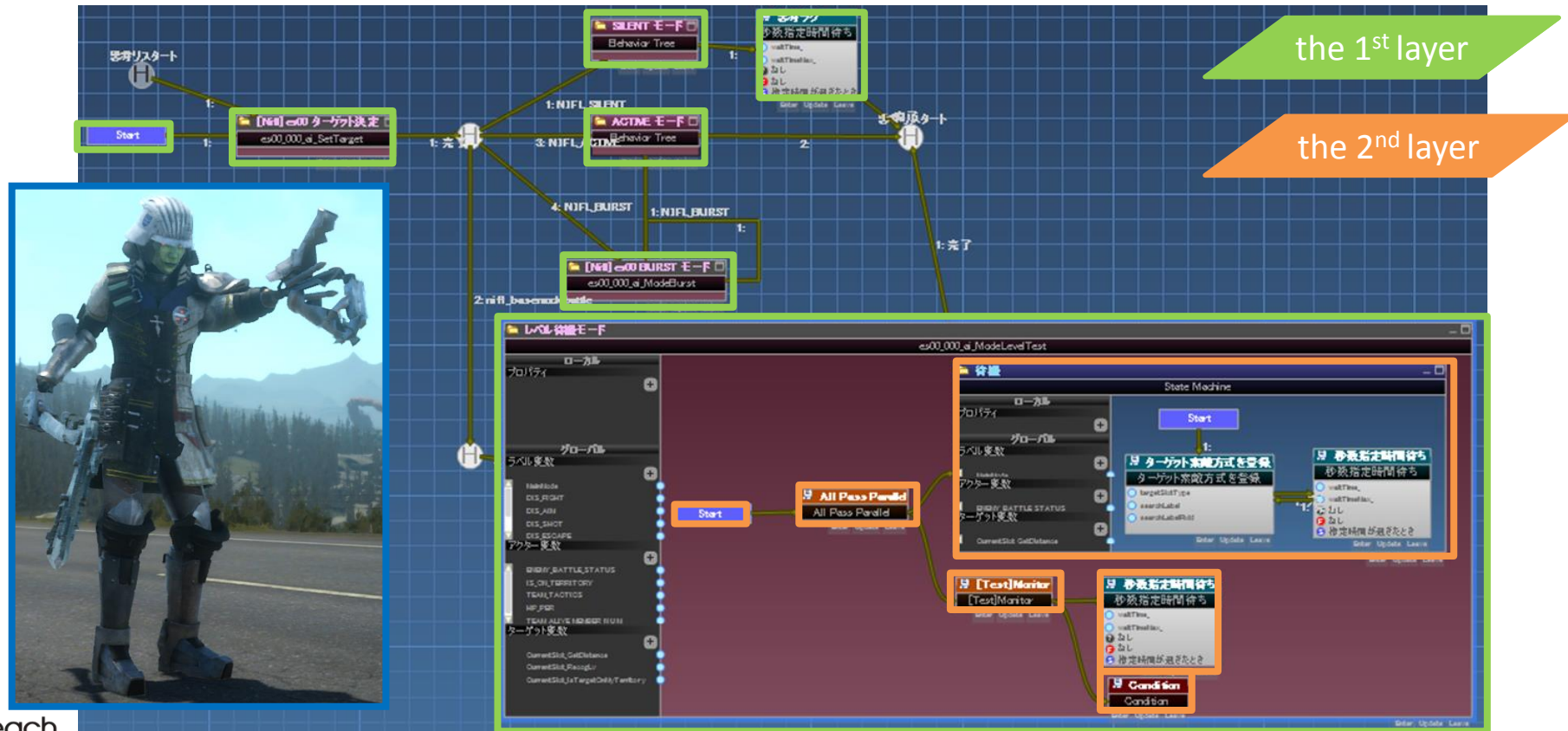
When one attacking state is added in lower layer, no change of top layer is required.



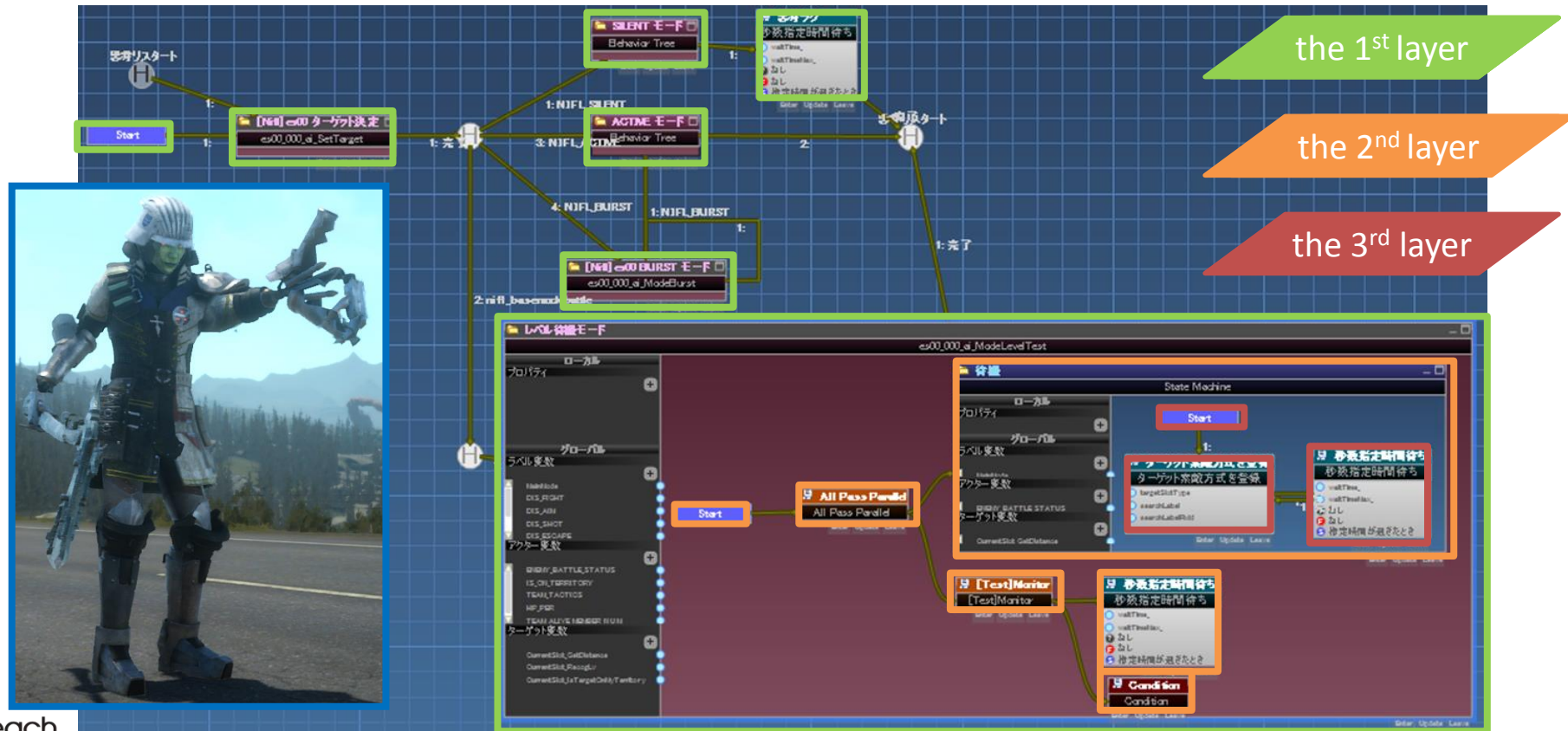
Scalability by Hierarchy



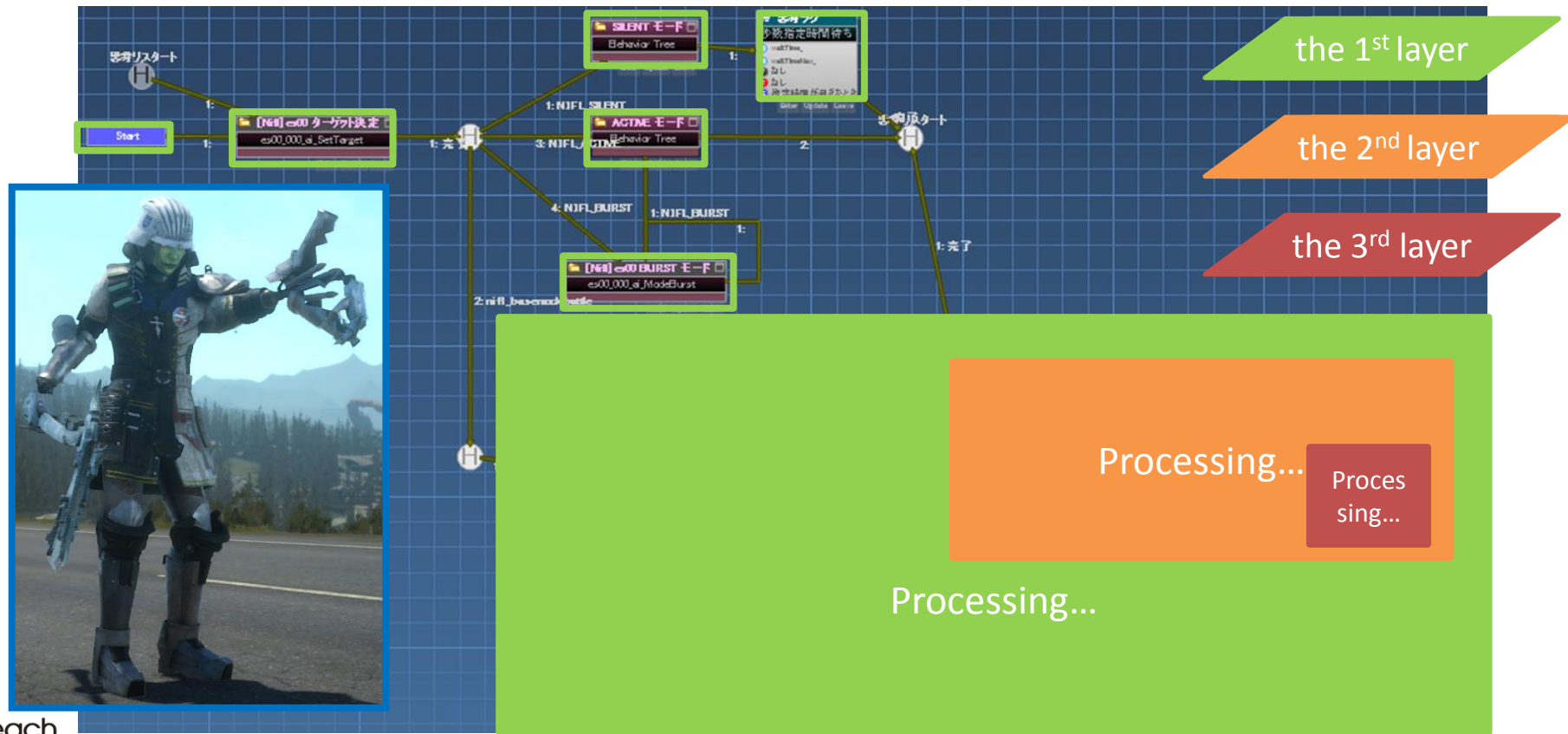
Scalability by Hierarchy



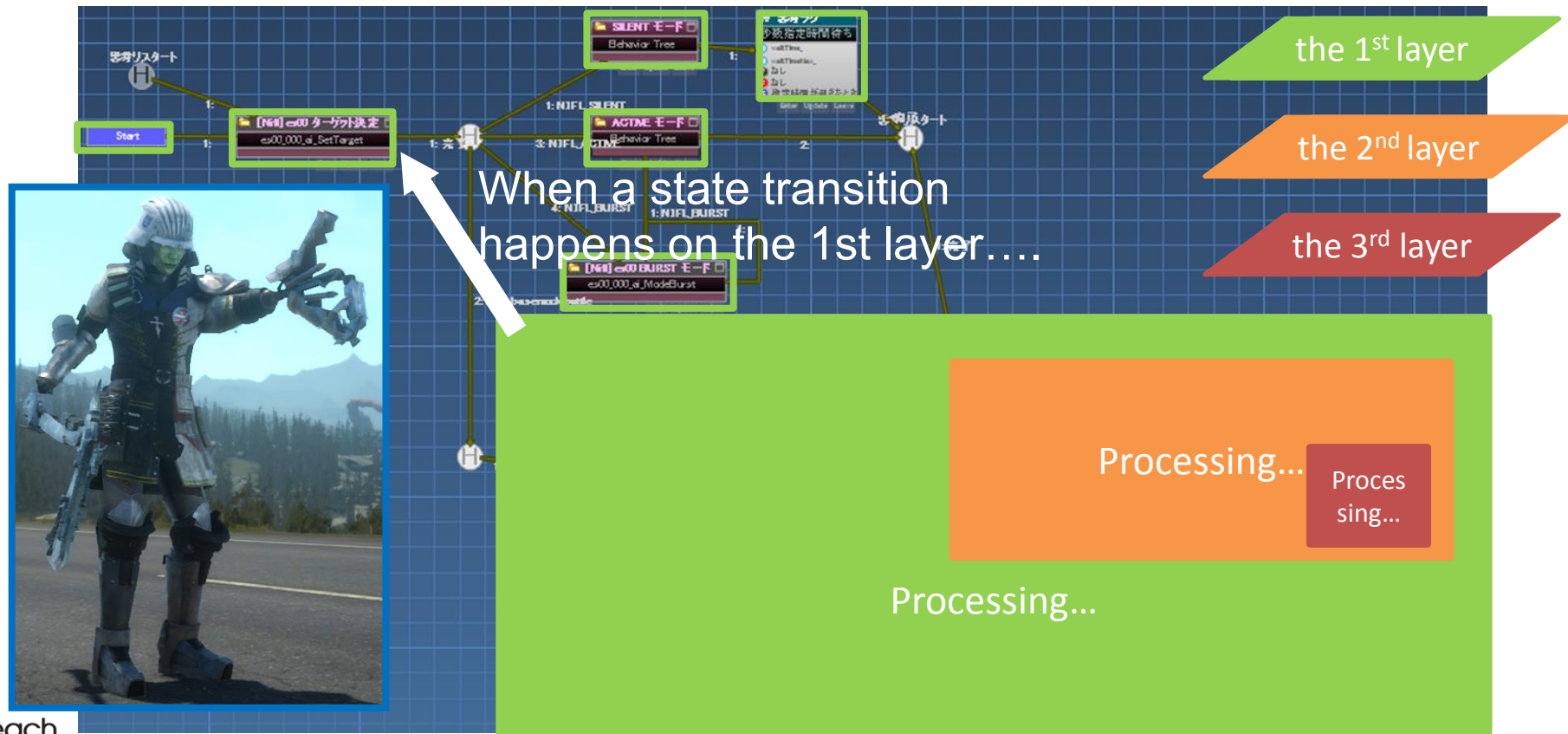
Scalability by Hierarchy



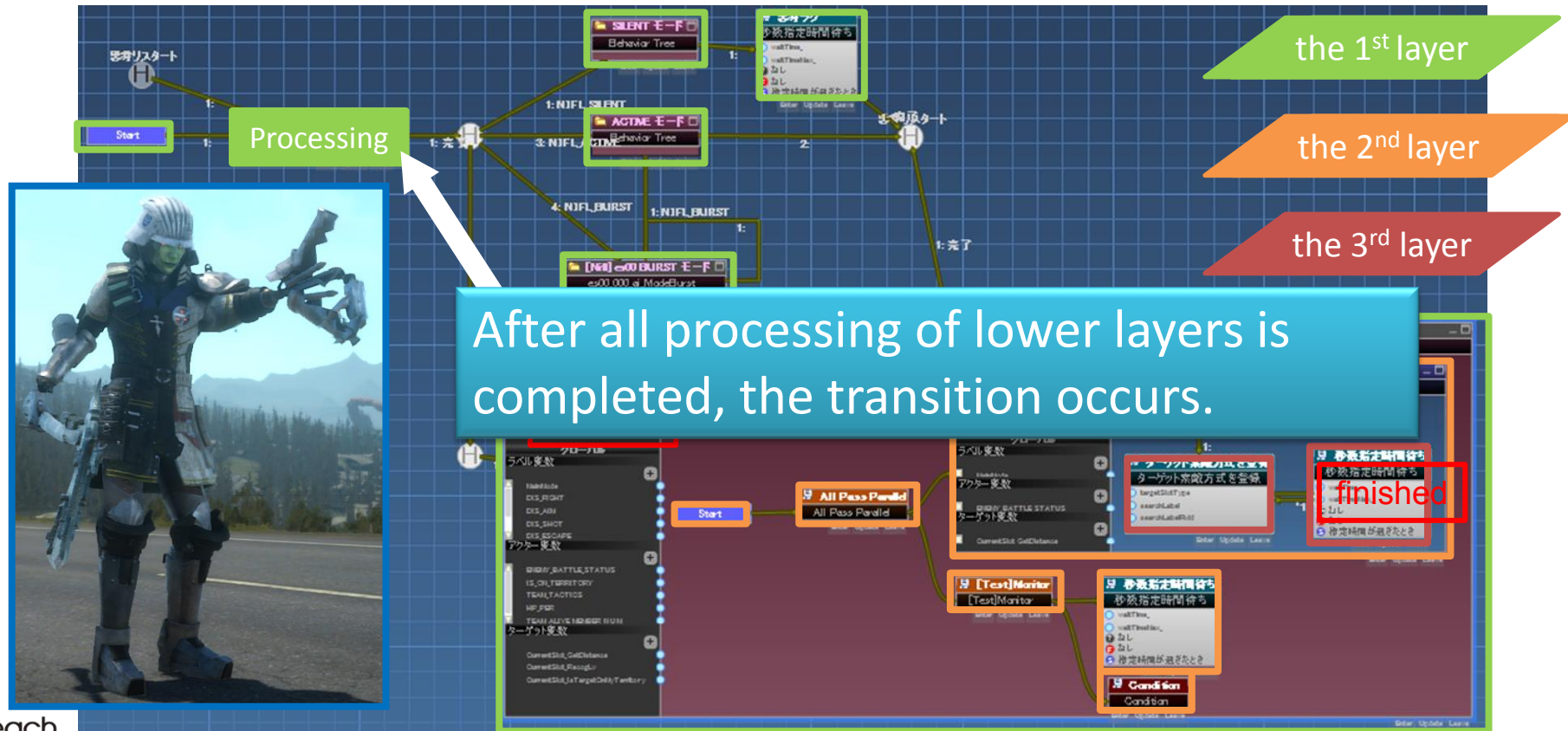
Scalability by Hierarchy



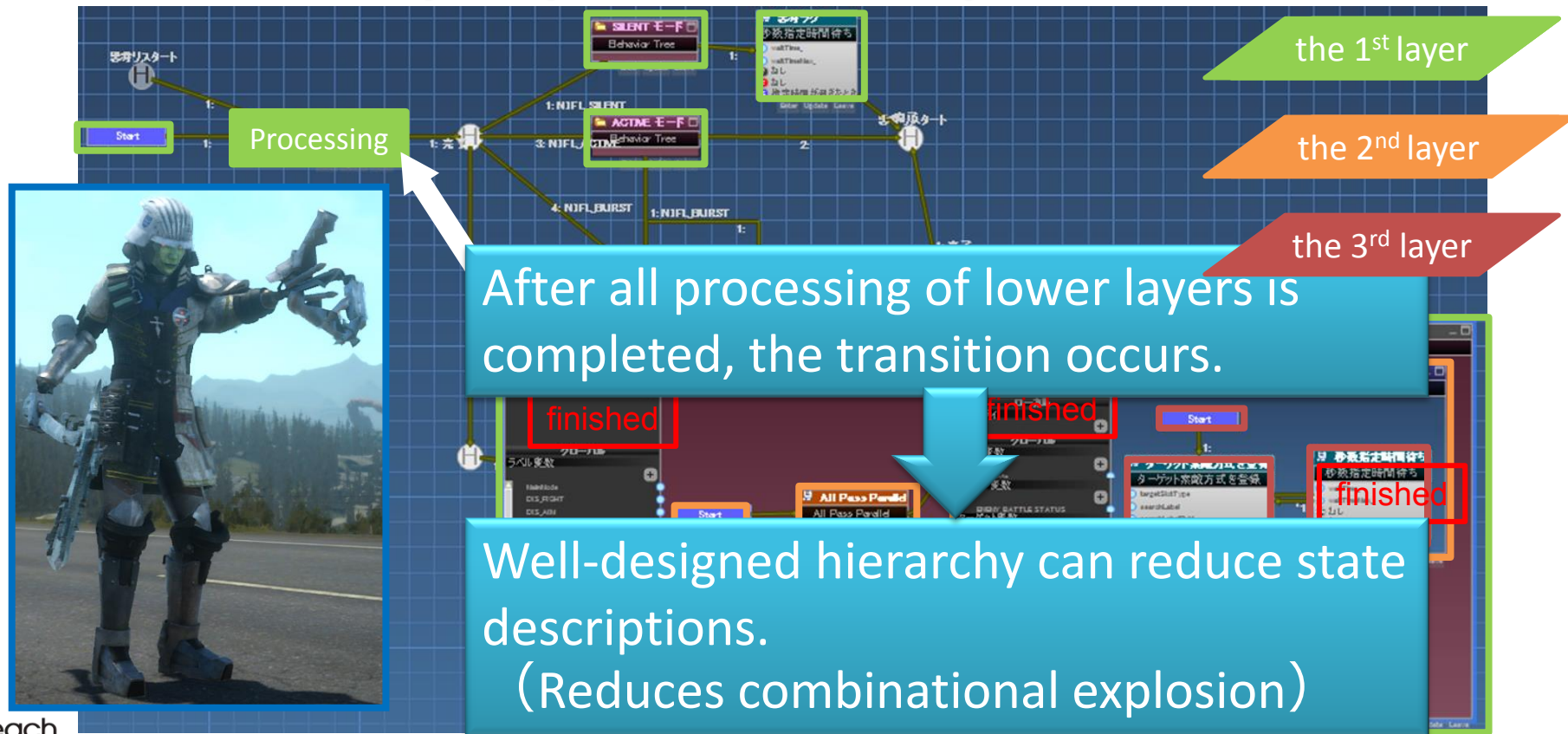
Scalability by Hierarchy



Scalability by Hierarchy



Scalability by Hierarchy



Problem 3

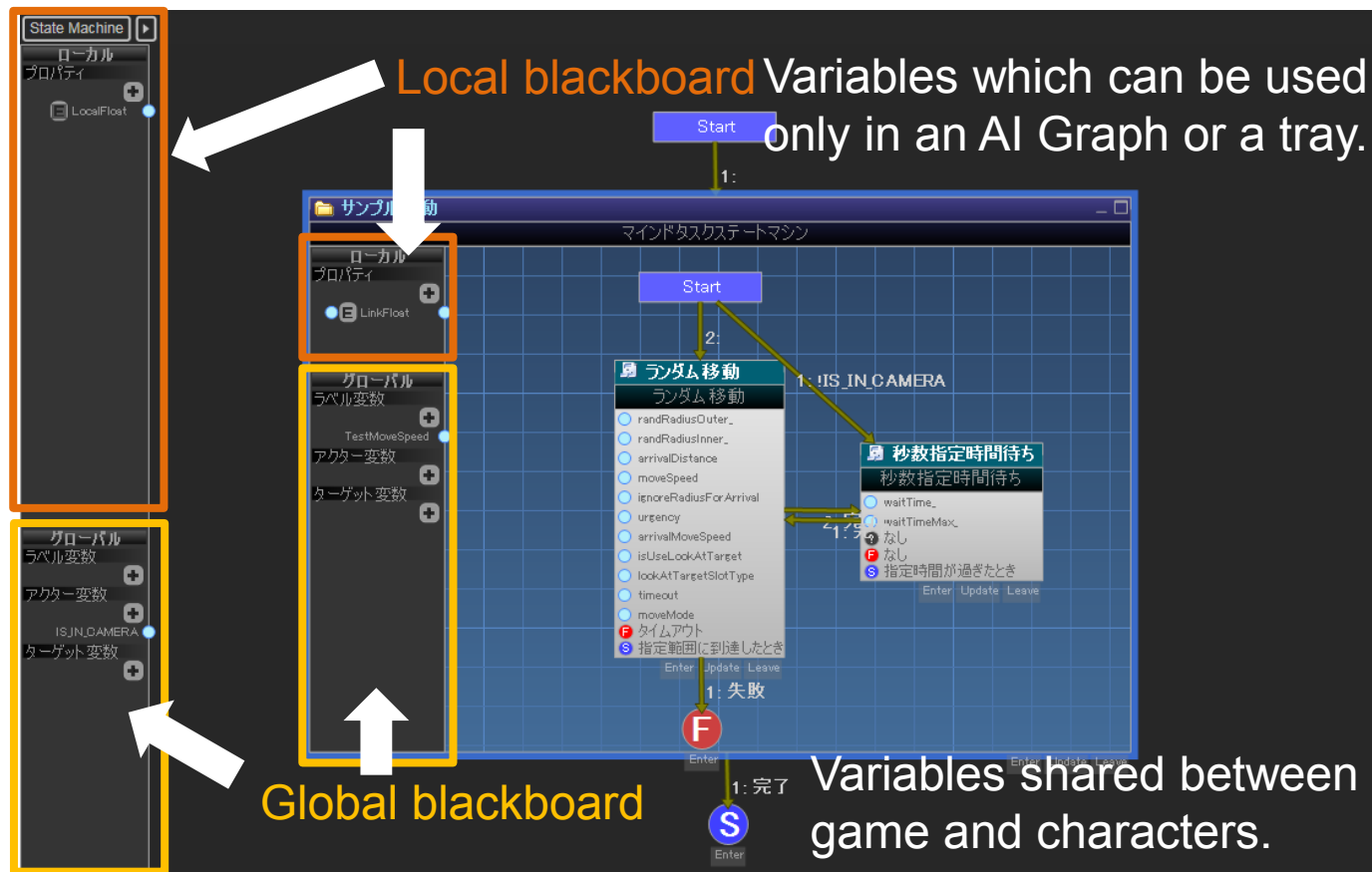


3. How to make smooth communication between game system and AI system ?

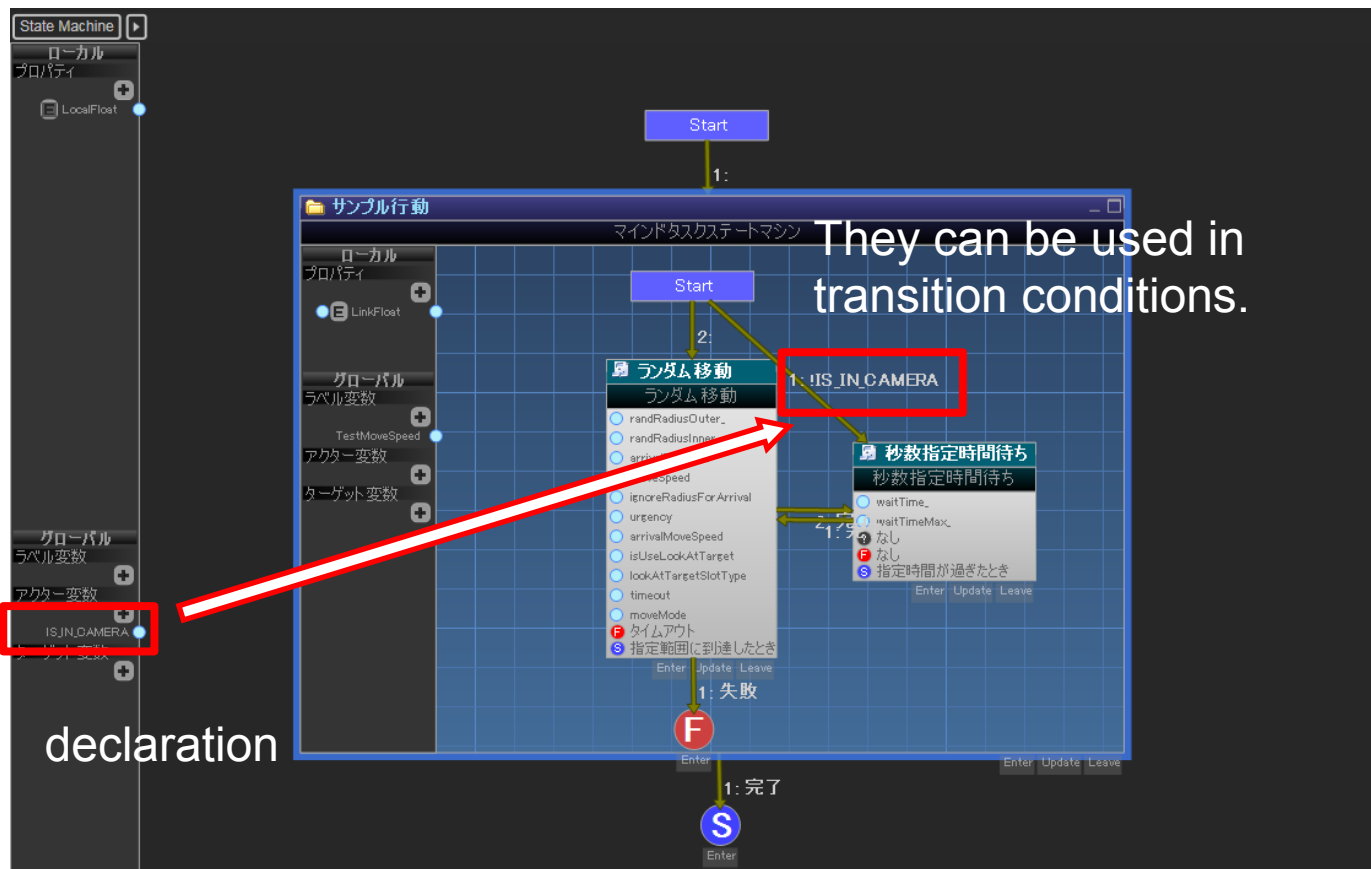
Game system and AI system can pass and get data smoothly via special region

⇒ 「**Blackboard**」

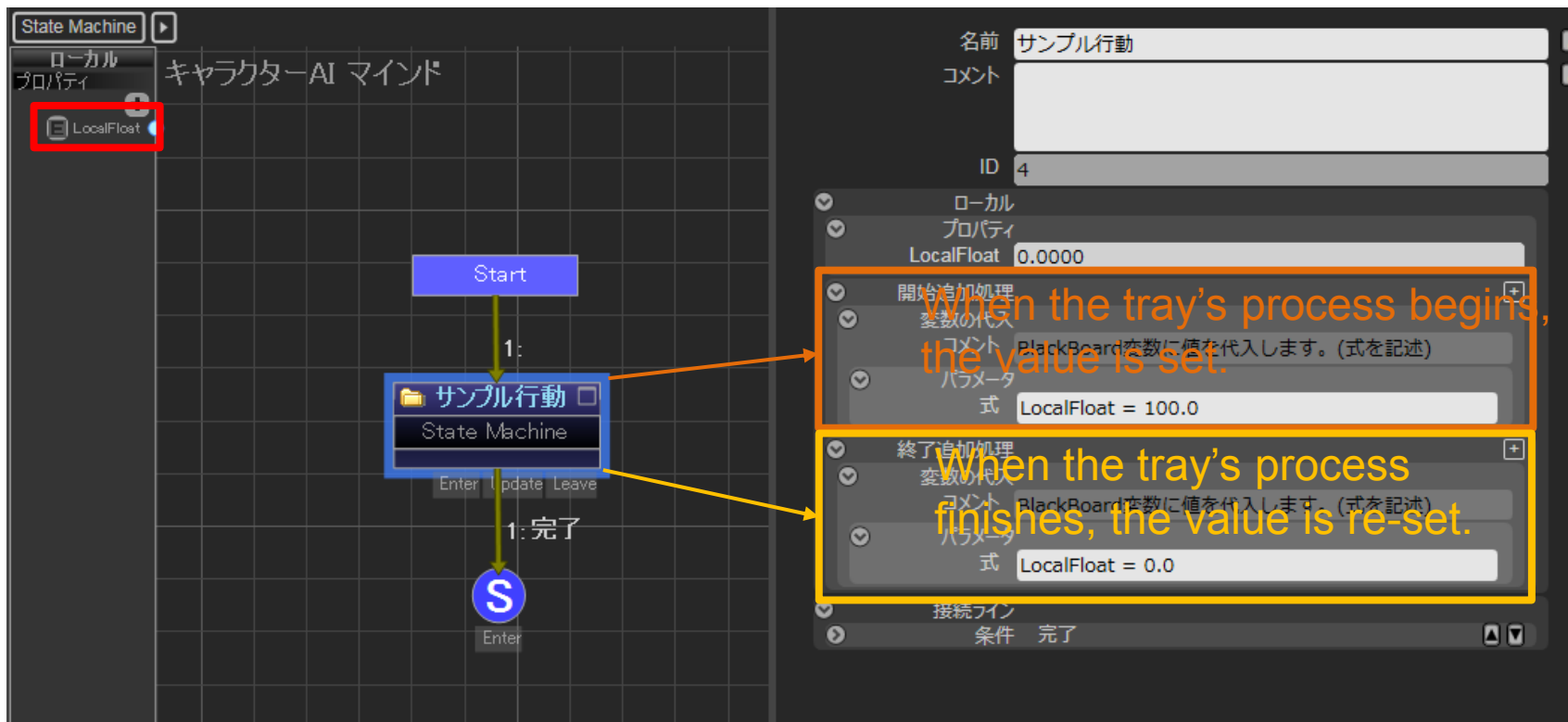
Blackboard



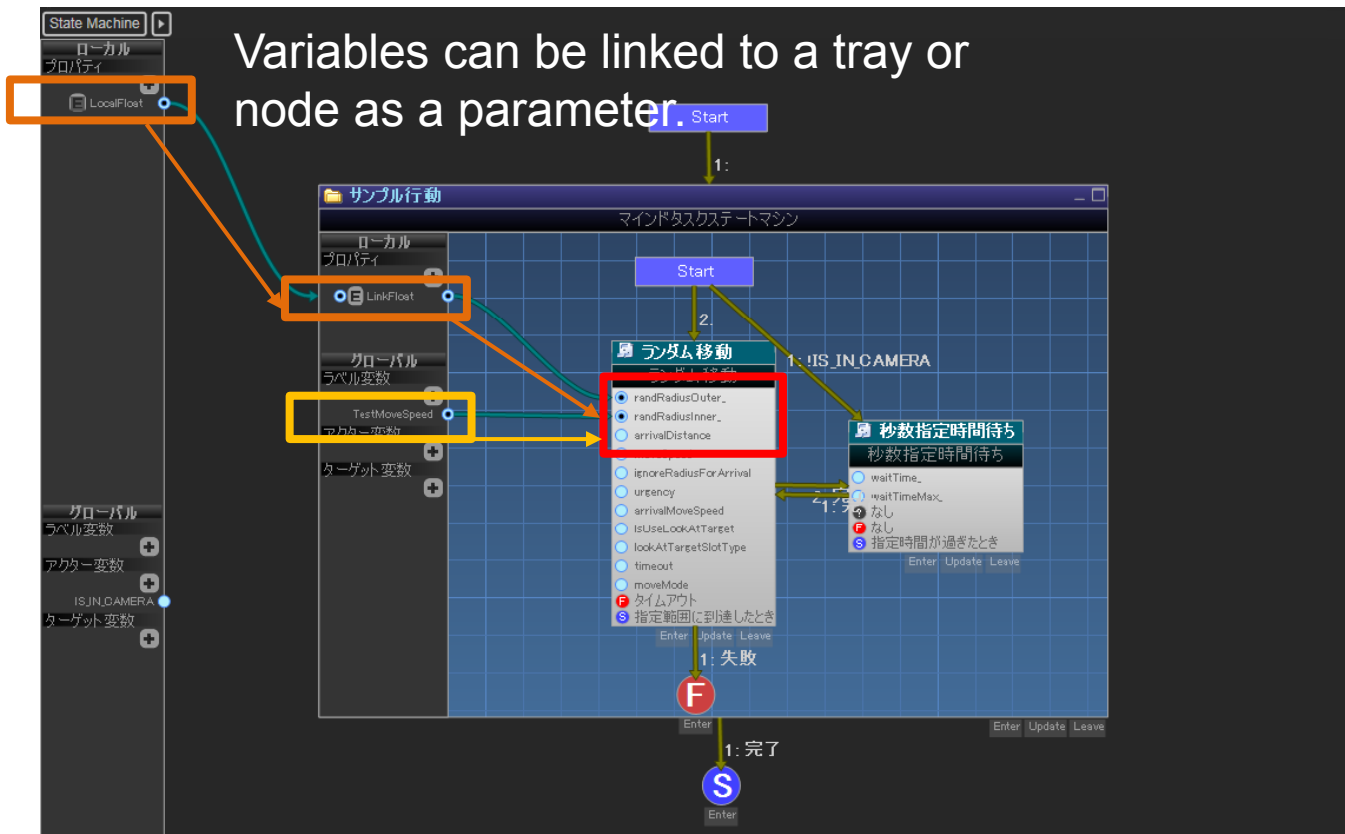
Blackboard



Blackboard



Blackboard



Problem 4

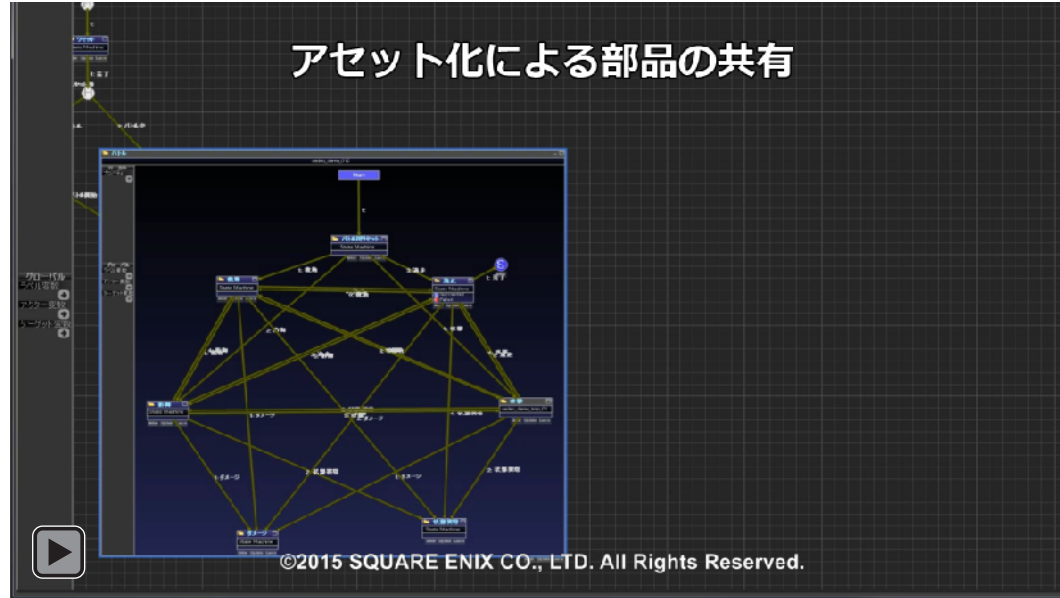


4. How to realize mass production of AI data without copy & paste ?

Any tray can be used as an **asset**, and its processing can be shared and re-used.

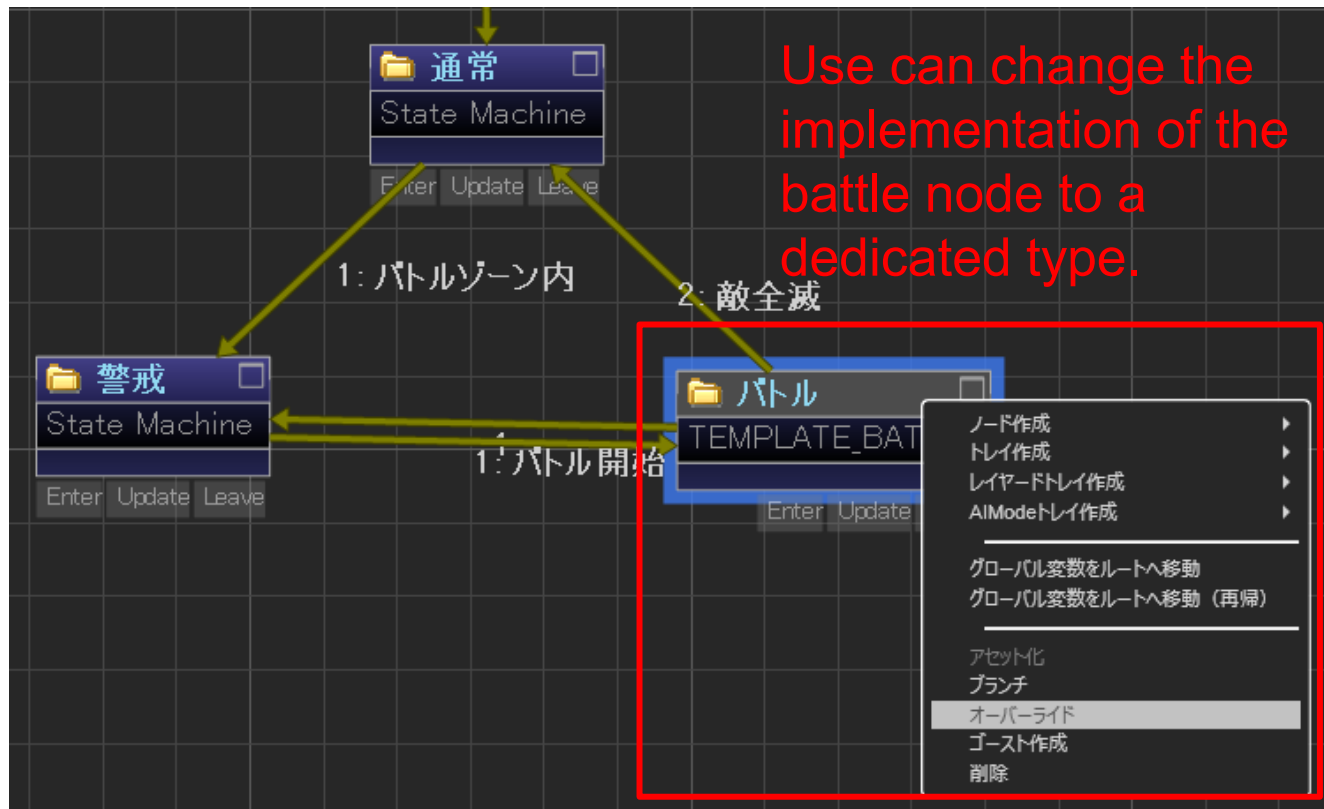
User can **override** a tray while sharing a top-layer structure.

Sharing a process by using an tray asset

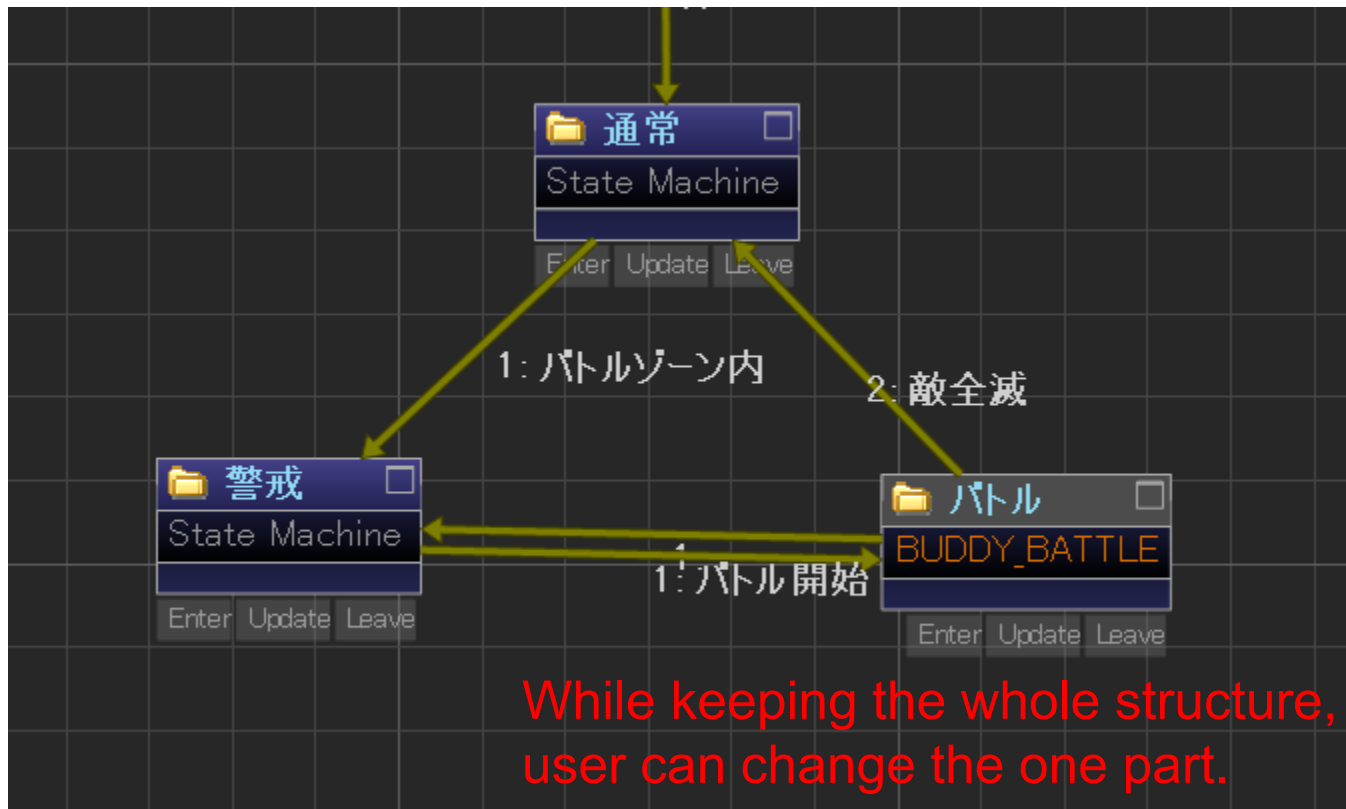


Asset-type Trays in different locations are processed simultaneously.

Sharing graph structure by override



Sharing graph structure by override



Summary



- Using a hybrid-node format, different graph systems are put together in a combined structure.
- Using a hierarchical data structure, scalability is ensured.
- Using a blackboard, the game and tools can simply exchange data through the UI
- Using conversion to Assets and overriding, we achieve mass production of AI without copy and pasting.

AI Graph Editor Design and Implementation



- 4 problems for tool design
- Some functions required by the game

Some functions required by the game



1. Must be able to confirm the execution status of the State and Behavior, also the value of the parameters and transition logs.
2. We want AI characters to perform seamlessly.
3. In order to seem more clever (ex. execute a new action while in the middle of another action), we want AI characters to always be thinking of something else.

Required function 1



1. Must be able to confirm the execution status of the State and Behavior, also the value of the parameters and transition logs.

Visual node debugger which can display this information in the tool, and

In-game debug window which displays detailed in-game logs.

Visual node debugger and In-game debug window



In-game debug window display detailed logs.

Visual node debugger displays current status of nodes.

Required function 2



2. We want AI characters to perform seamlessly.

User can interrupt AI's logic by making the prepared tray active from the outside of system.

⇒ 「Tray interruption processing」

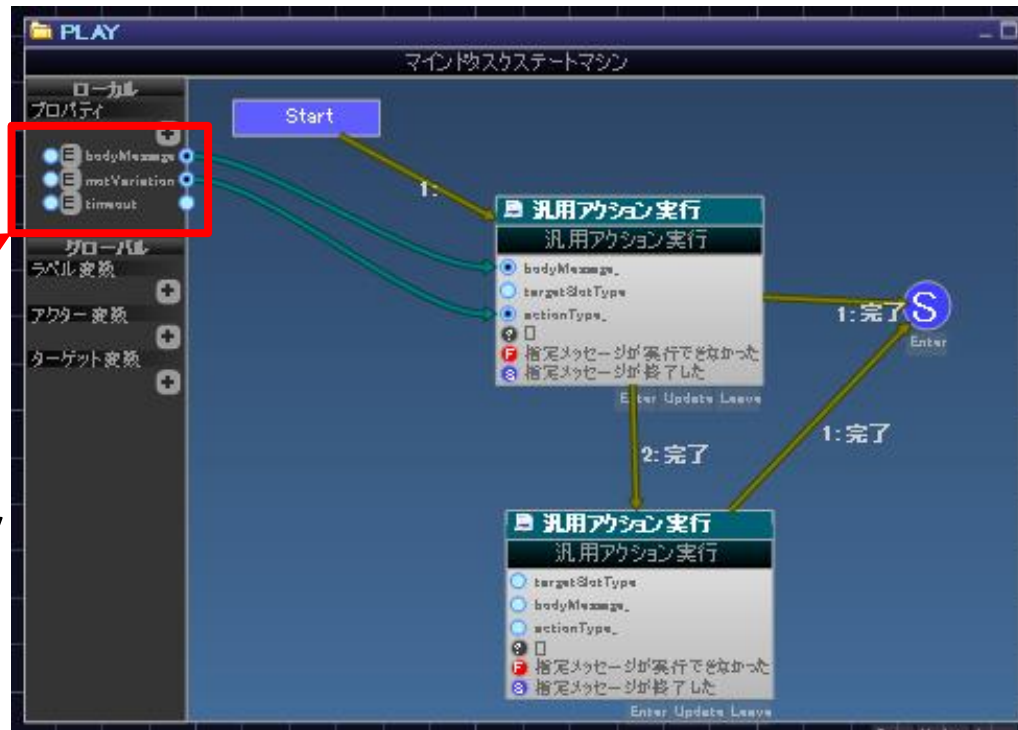
Tray interruption processing

Preparing a tray where user described simple action of AI.

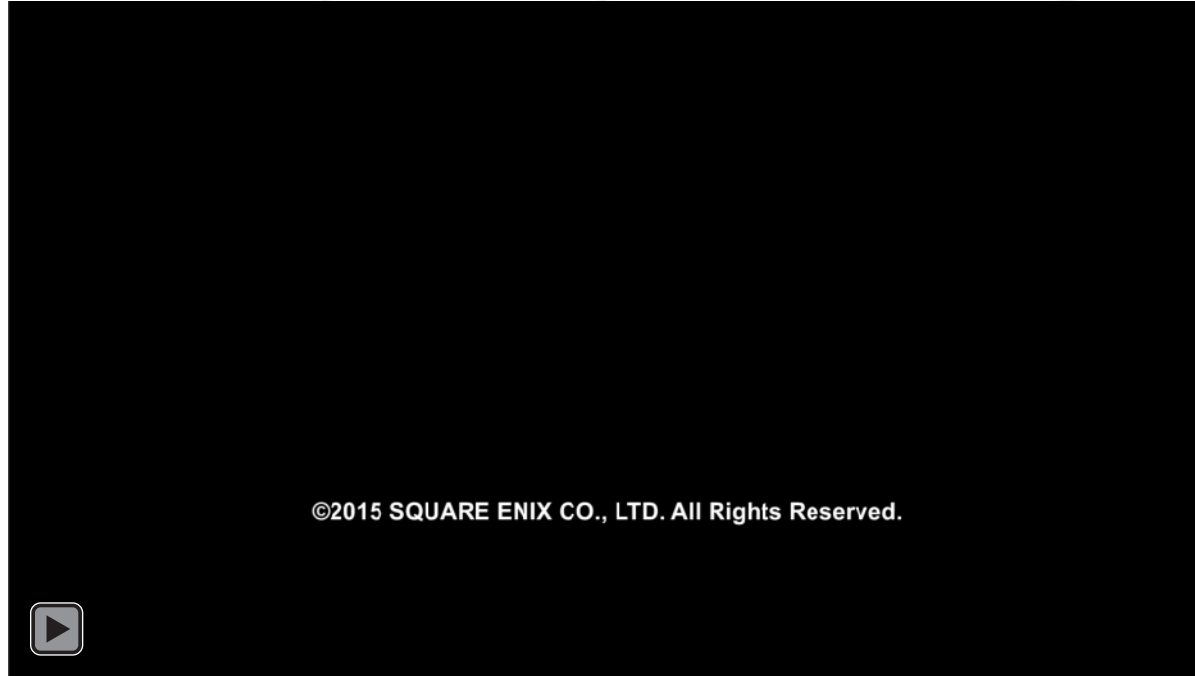
(The other file is OK.)

Passing parameters via blackboard.

After the process of the tray finishes, the process turns back to the original system automatically.



Tray interruption processing



After the event (running monsters appear rapidly to go to the player's position) happens, they turn back to regular AI mode.

Required function 3

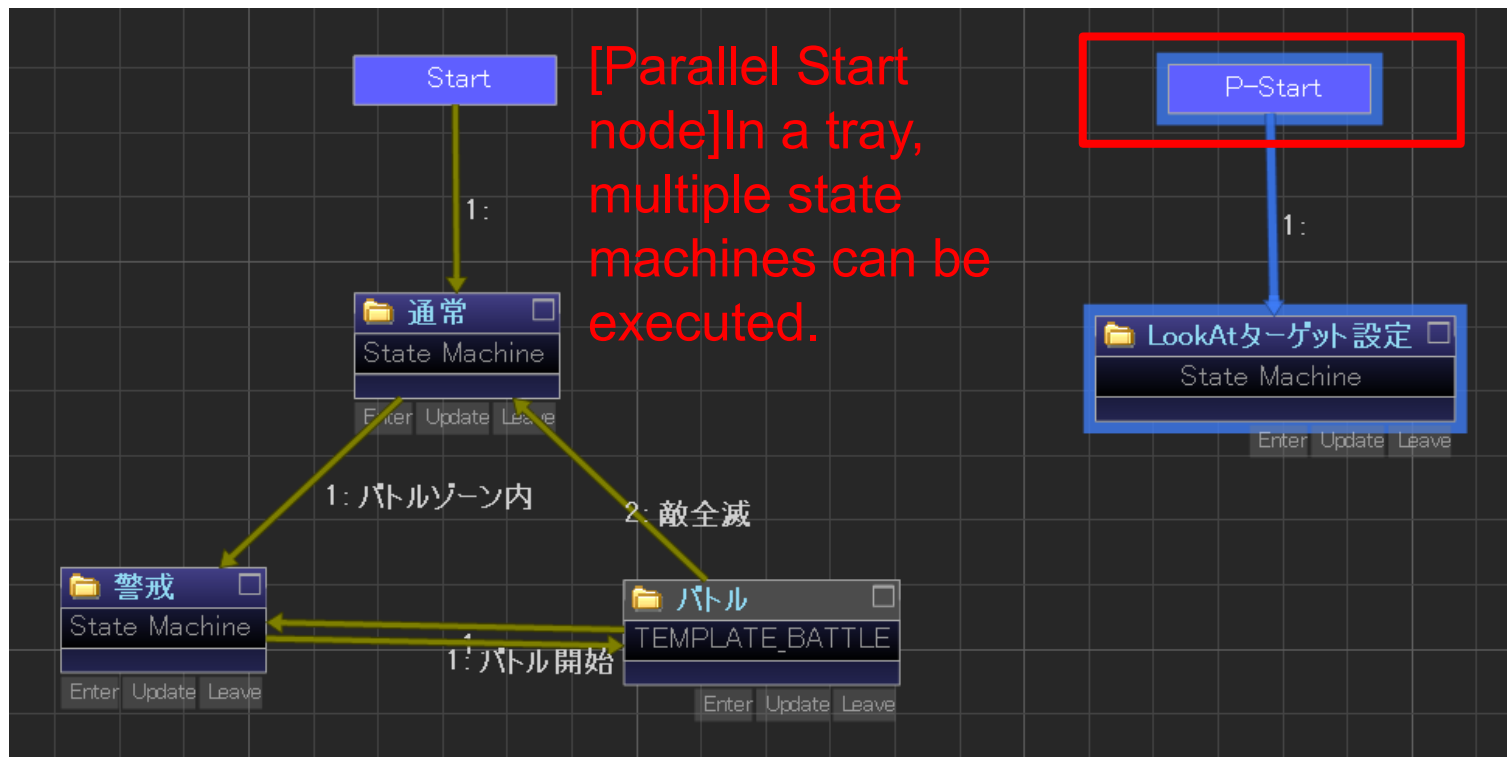


3. In order to seem more clever (ex. execute a new action while in the middle of another action), we want AI characters to always be thinking of something else.

Parallel thinking is introduced both for state machine and behavior tree.

Parallel thinking

- State machine case



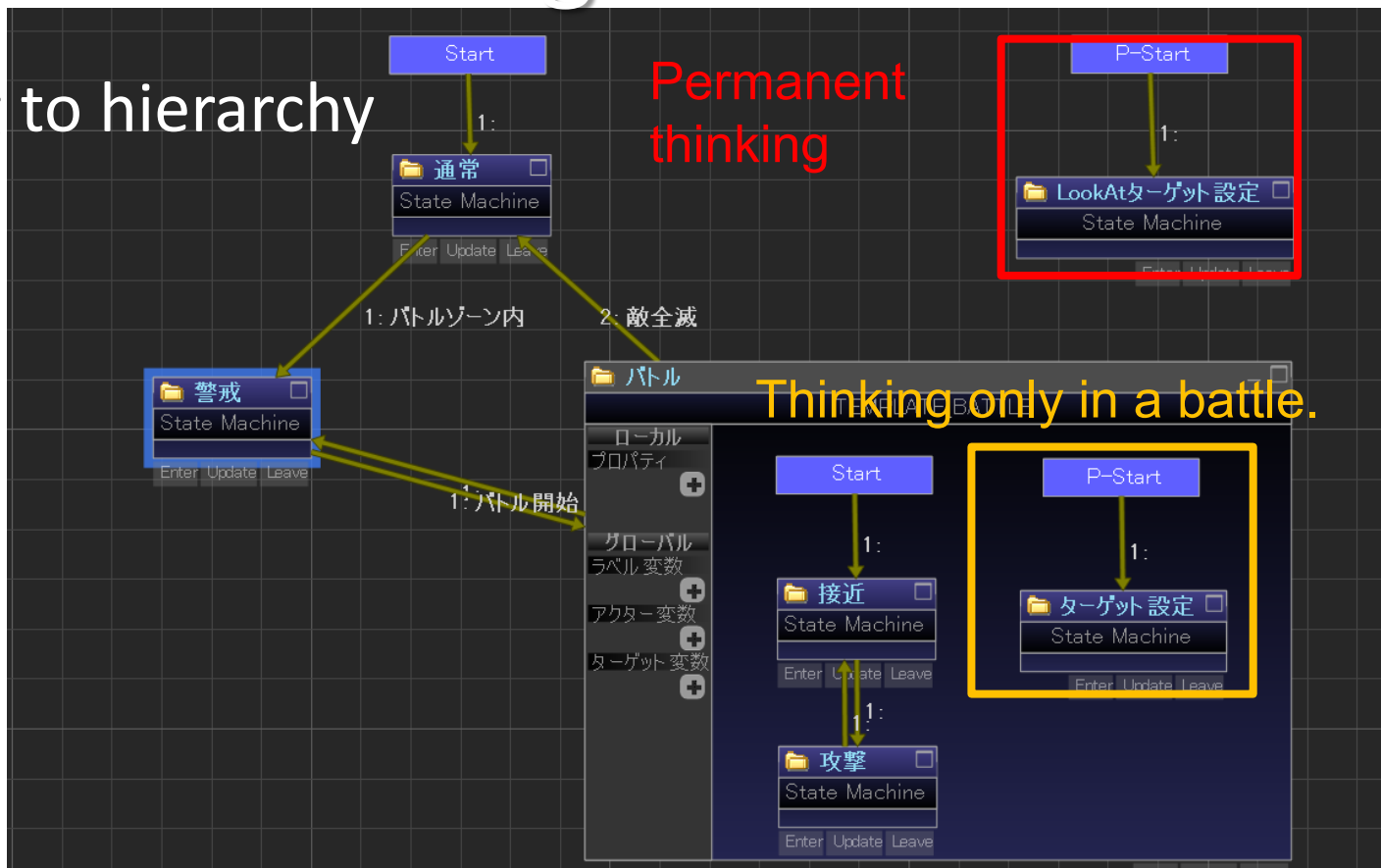
Parallel thinking

- Behavior tree case



Parallel thinking

Applying to hierarchy



Parallel thinking



Parallel thinking on a behavior tree.

©2015 SQUARE ENIX CO., LTD. All Rights Reserved.

Summary



- Visual debugger enables real-time watching of current status.
- Tray interruption processing can realize a seamless transition to an acting of event.
- Parallel thinking makes AI more smart.