Persistent, Multi-player Games with Rules

- Andrew Waterman, M.A.,
  Técnico Académico,
  Titular B,
  ECOSUR SCLC

  awaterma@ecosur.mx

- Max Pimm

  max@alwayssunny.com

Our Approach

- Allow participation on the Network in a standards-based, scalable fashion

- Allow offline analysis of user actions through any database and data-mining tool

- Allow researchers to work with logical rules, not code

- Provide an open scripting framework for modeling and user interaction

Our Approach

- Concurrent, networked system

  - Based upon widely accepted and globally available open-source frameworks

- Use of the industry developed standards (EJB3, MVC, JMS, JSR 94 [Rules engine], JSR 223 [Scripting framework])

Implementation

- Stack is segregated into 4 discrete layers

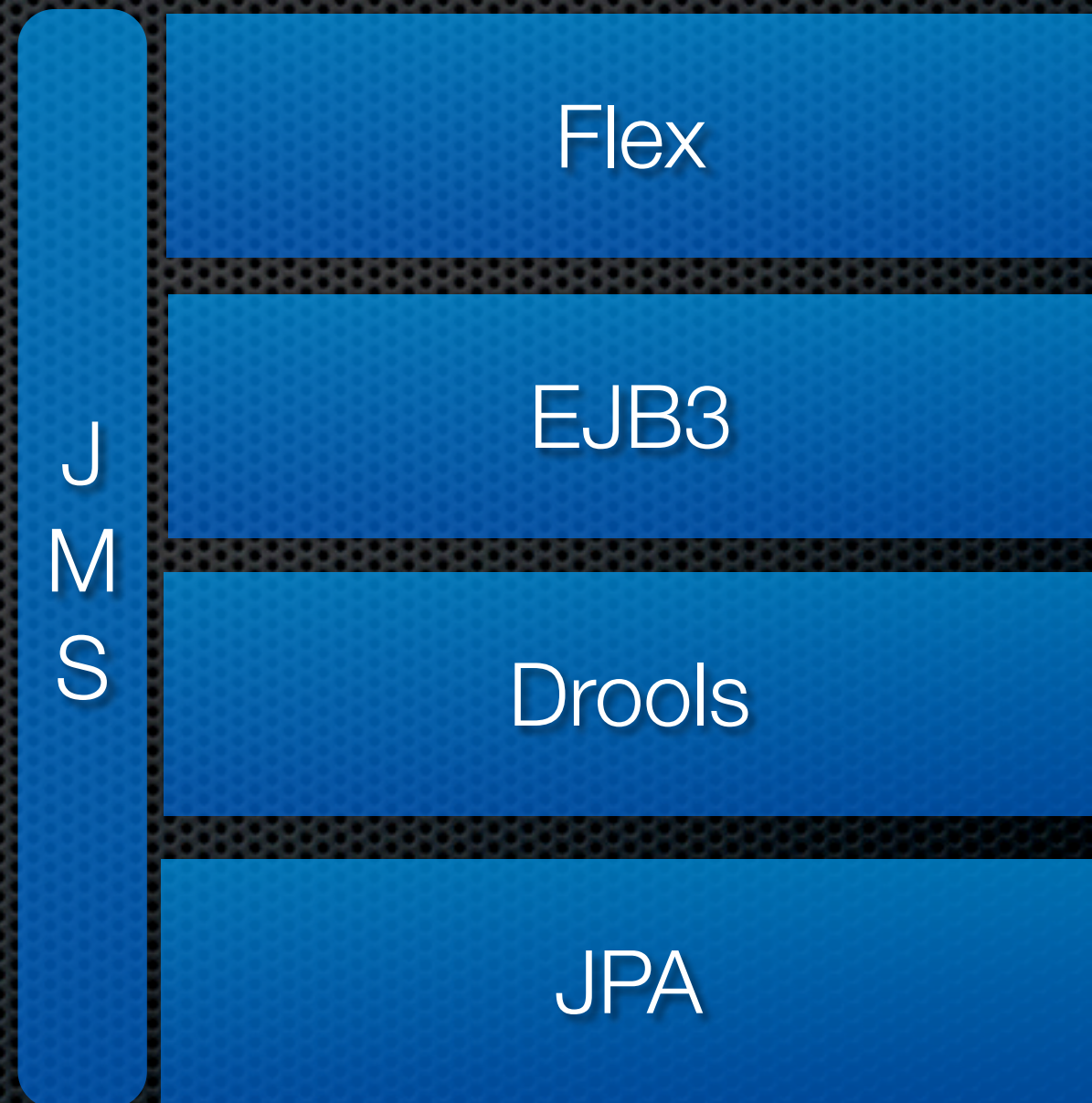- Allows experts to concentrate upon their domain

| Display |
| Modeling |
| Logic |
| Persistence |

Prototype

- Flex3 for user interface (display)

- EJB3 for concurrency/ modeling

- JMS for asynchronous messaging

- Drools rules engine for game logic

- JPA for persistancy

J M S	Flex
	EJB3
	Drools
	JPA

GENTE

- Luis Garcia Barrios has developed "GENTE", a cooperation dilemma game derived from Pente, for use in our prototype

- Custom UI and modeling layers built on top of common infrastructure

- Game expressed in logical rules, models in code

GENTE (in-play)

GENTE Rules

- Uses a 19x19 "Go" board

- 4 players, divided into 2 teams

- 10 points awarded per game

- Yellow must go first, and move to the center square

- Players are not allowed to capture the opposing team's tokens

- 2 pairs of three of one color, that player awarded all 10 points

- 3 pairs of four of team colors, both players awarded 5 points

Demo

Example

```
   rule "score"
    agenda-group "evaluate"
    salience 2
    no-loop true
when
    game   : PenteGame (state == GameState.PLAY)
    move   : PenteMove (player.turn == true, status == Status.MOVED)
then
    int score = scorePlayer (move);
    PentePlayer player = (PentePlayer) move.getPlayer ();
    if (score == 5) {
        PentePlayer teammate = getPartner (player);
        teammate.setPoints (score);
    }
    player.setPoints (score);
    modify (move) { setPlayer (player) }
end
```

Under the covers

Move

↓

Rule

↓
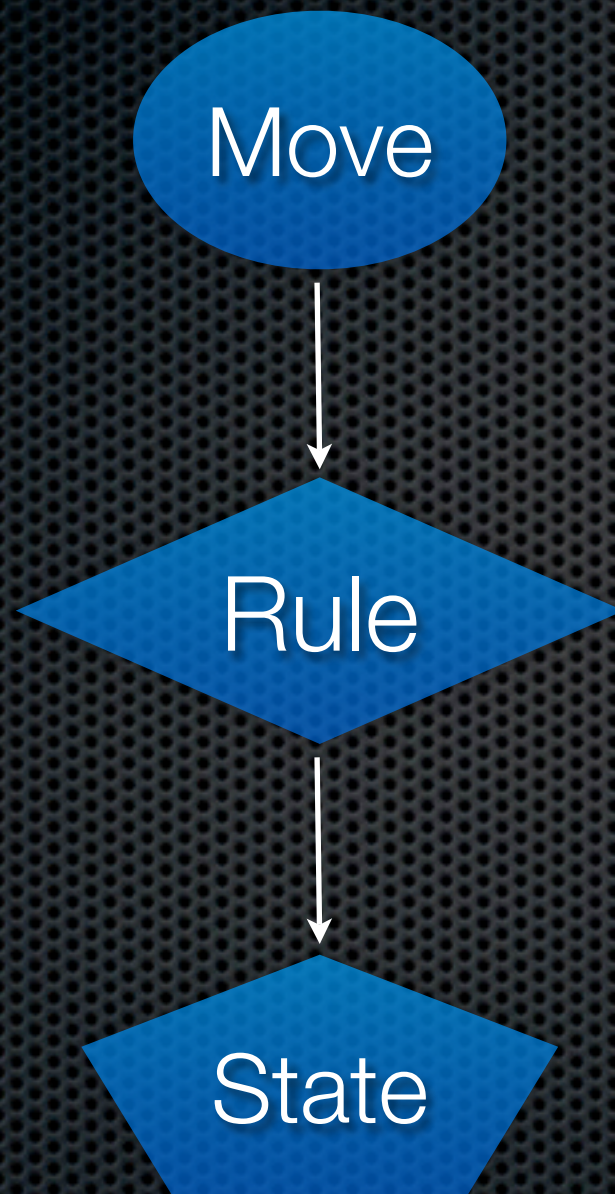
State

- Player makes a move

- Move is evaluated by the rules

- Rules change persistent state (when valid)

Drools Rule Engine

* Game logic is expressed as rules in Drools Rule Language (DRL)

  * mix of Java syntax with simple, basic logic

* Plain language expressions (English, Spanish) can be mapped to rules DRL

* Rules can be compiled in other formats, besides DRL (XML, Clips)

Future

* Looking at JSR 223 (Java's open scripting framework) to allow different scripting languages to plug-in to the rules, modeling and concurrency layers; allowing quicker implementation of interactive simulations by researchers and software engineers

* Exploring strategy based agents

* Exploring agents that can learn from participant behaviors and strategies

Questions