

# HANGMAN

?

II



A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P
Q	R	S	T
U	V	W	X
Y	Z		

? ? ? ?

Bibek Pandey - Elixir Hangman Portfolio Deck

CLI

--name (player id) --type ("human" or "robot") --random (num random secrets, max 1000) [--secret (hangman word(s)) --baseline] [--log --display -timeout] [--parallel (suggested w/40 secrets or more)]

or aliases: -n (player id) -t ("human" or "robot") -r (num random secrets, max 1000) [-s (hangman word(s)) -bl] [-l -d -ti] [-pl]

```
$ mix escript.build
```

```
$ ./hangman_game -n martha -t robot -r 1 -d
```

```
#martha_feed --> Game 1 has started
```

```
#martha_feed Game 1, secret length --> 10
```

```
#martha_feed Game 1, letter --> e
```

```
#martha_feed Game 1, Round 1, status --> -----; score=1;
status=KEEP_GUESSING
```

```
#martha_feed Game 1, letter --> i
```

```
#martha_feed Game 1, Round 2, status --> -----I--I-; score=2;
status=KEEP_GUESSING
```

```
#martha_feed Game 1, letter --> s
```

```
#martha_feed Game 1, Round 3, status --> -----IS-I-; score=3;
status=KEEP_GUESSING
```

```
#martha_feed Game 1, letter --> t
```

```
#martha_feed Game 1, Round 4, status --> -----ISTI-; score=4;
status=KEEP_GUESSING
```

```
#martha_feed Game 1, letter --> a
```

```
#martha_feed Game 1, Round 5, status --> ----A-ISTI-; score=5;
status=KEEP_GUESSING
```

```
#martha_feed Game 1, letter --> r
```

```
#martha_feed Game 1, Round 6, status --> ----A-ISTI-; score=6;
status=KEEP_GUESSING
```

```
#martha_feed Game 1, letter --> d
```

```
#martha_feed Game 1, Round 7, status --> D--A-ISTI-; score=7;
status=KEEP_GUESSING
```

```
#martha_feed Game 1, word --> dynamic
```

```
#martha_feed Game 1, Round 8, status --> DYNAMIC; score=7; status=GAME_WON
```

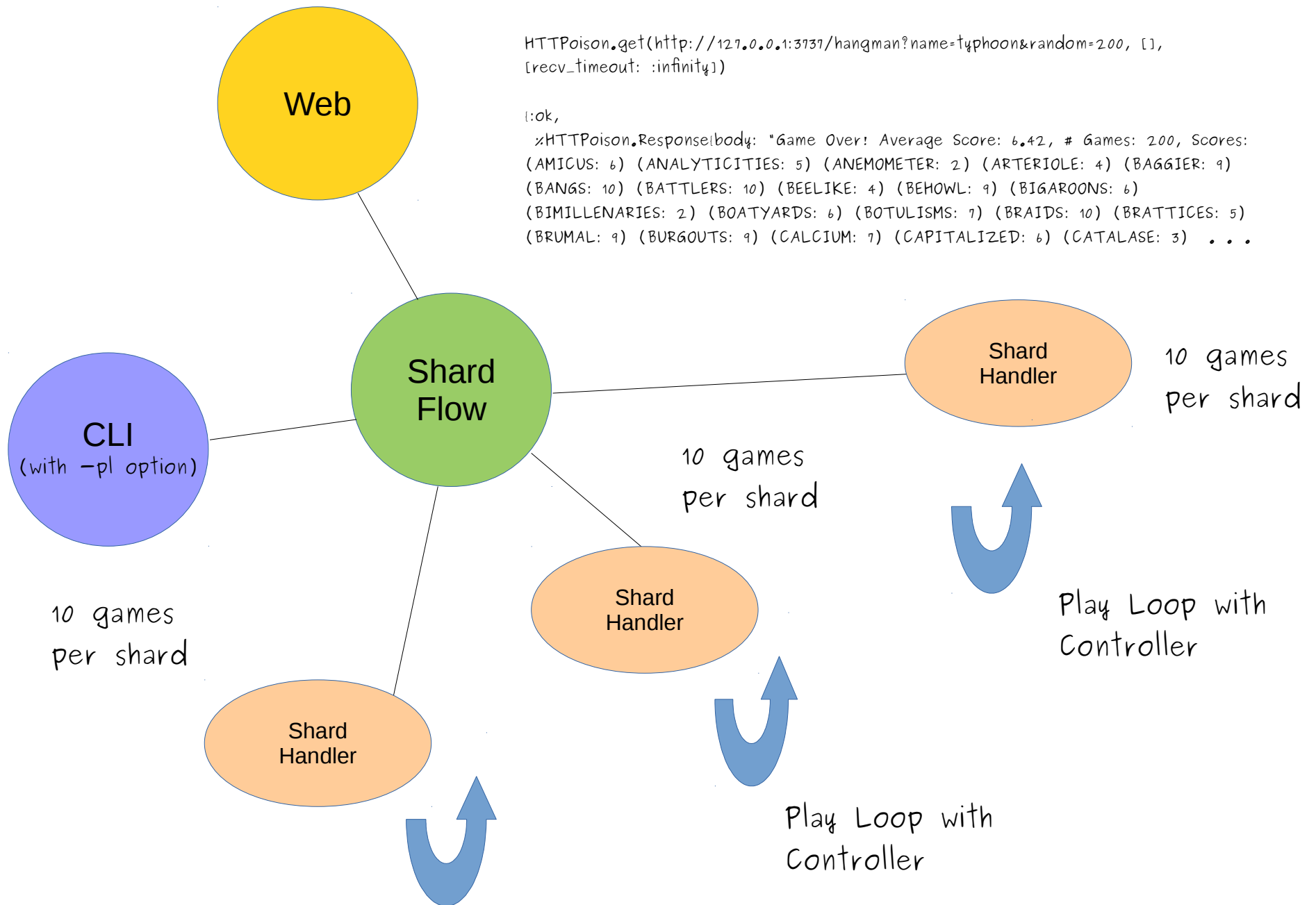
```
#martha_feed Game Over!! --> Game Over! Average Score: 7.0, # Games: 1, Scores:
(DYNAMIC: 7)
```

Handler



Play Loop with  
Controller

## Hangman Command Line



```
HTTPOison.get(http://127.0.0.1:3737/hangman?name=typhoon&random=200, [],
[recv_timeout: :infinity])
```

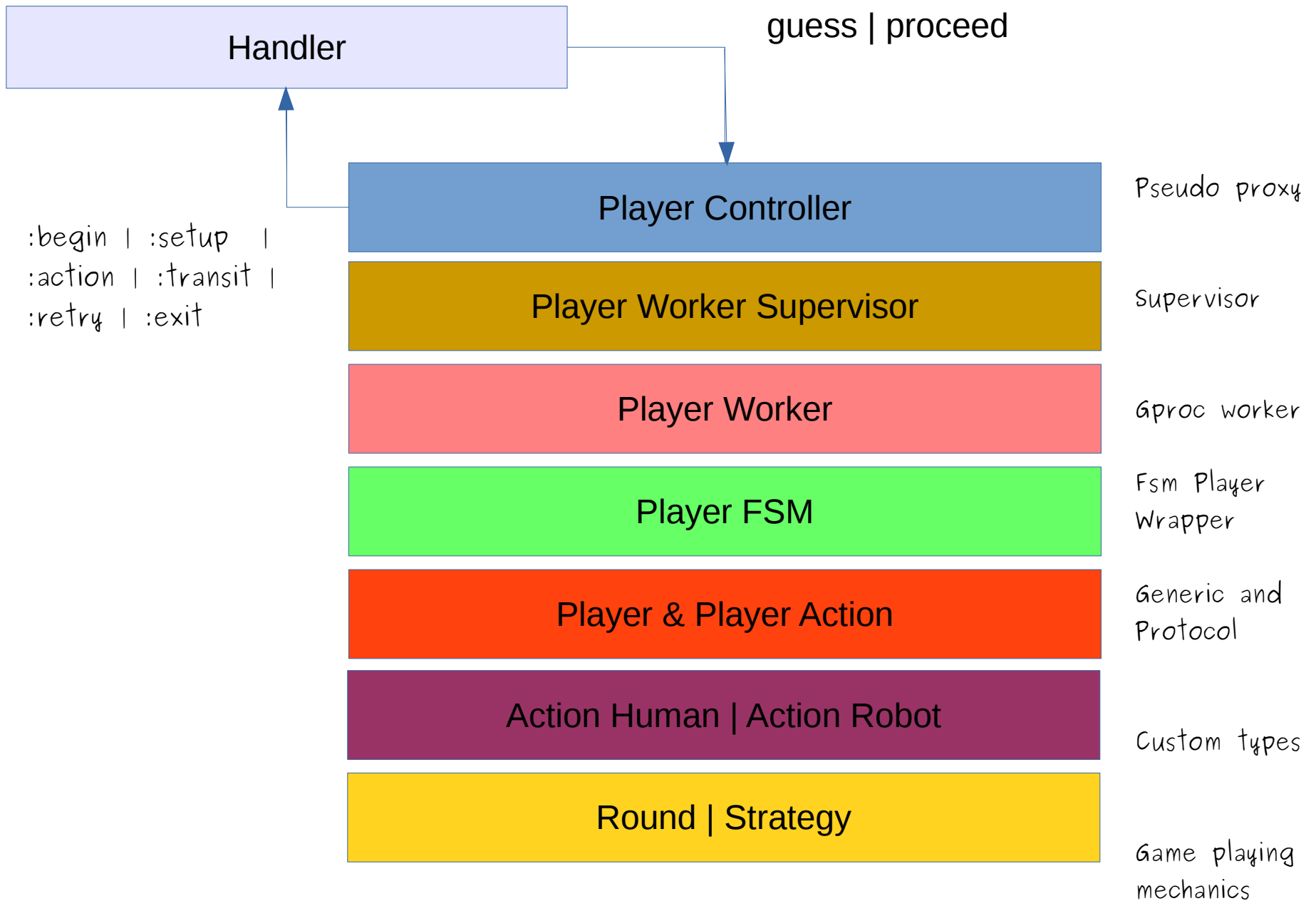
```
!ok,
```

```
⌘HTTPOison.Response{body: "Game Over: Average Score: 6.42, # Games: 200, Scores:
(AMICUS: 6) (ANALYTICITIES: 5) (ANEMOMETER: 2) (ARTERIOLE: 4) (BAGGIER: 9)
(BANGS: 10) (BATTLERS: 10) (BEELIKE: 4) (BEHOWL: 9) (BIGAROONS: 6)
(BIMILLENARIES: 2) (BOATYARDS: 6) (BOTULISMS: 7) (BRAIDS: 10) (BRATTICES: 5)
(BRUMAL: 9) (BURGOUTS: 9) (CALCIUM: 7) (CAPITALIZED: 6) (CATALASE: 3) . . .
```

```
HTTPOison.get http://127.0.0.1:3737/hangman?
name=typhoon&secret[]=eel&secret[]=porcupine
```

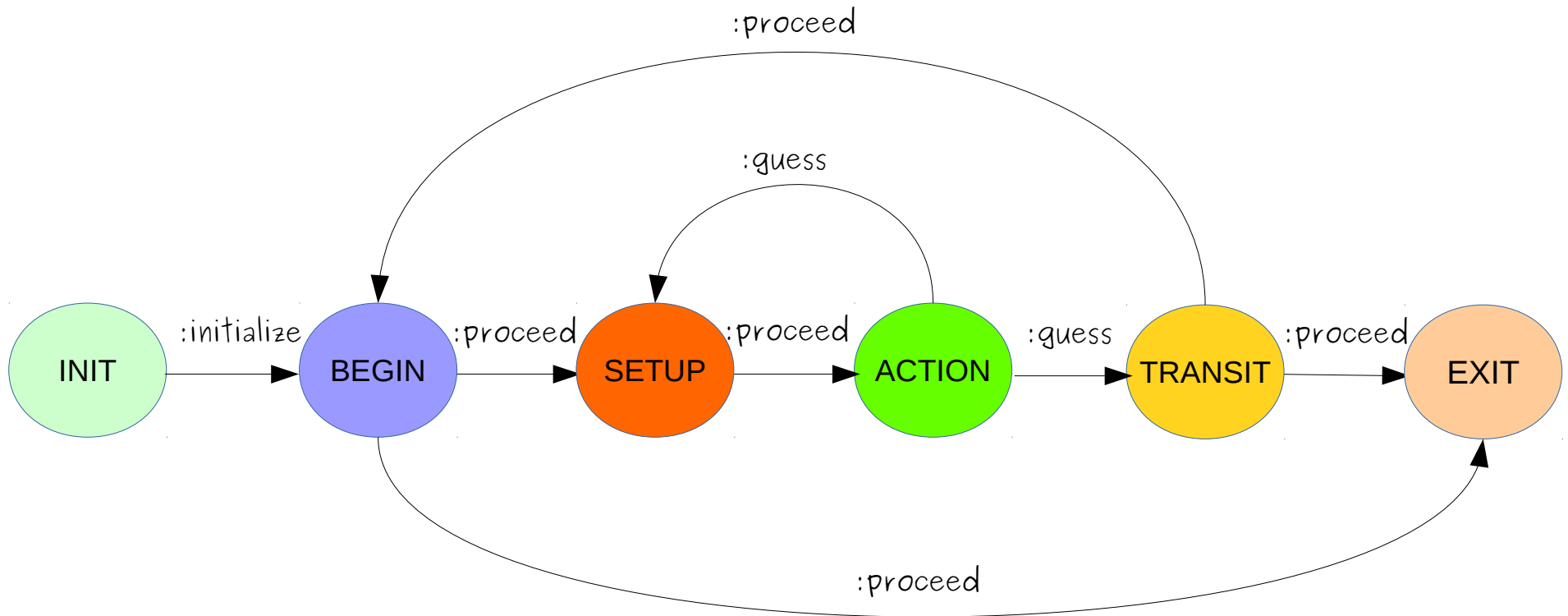
Play Loop with  
Controller

## Parallel Hangman



Player Abstraction Stack

# Player Finite State Machine Flow



init - Start state

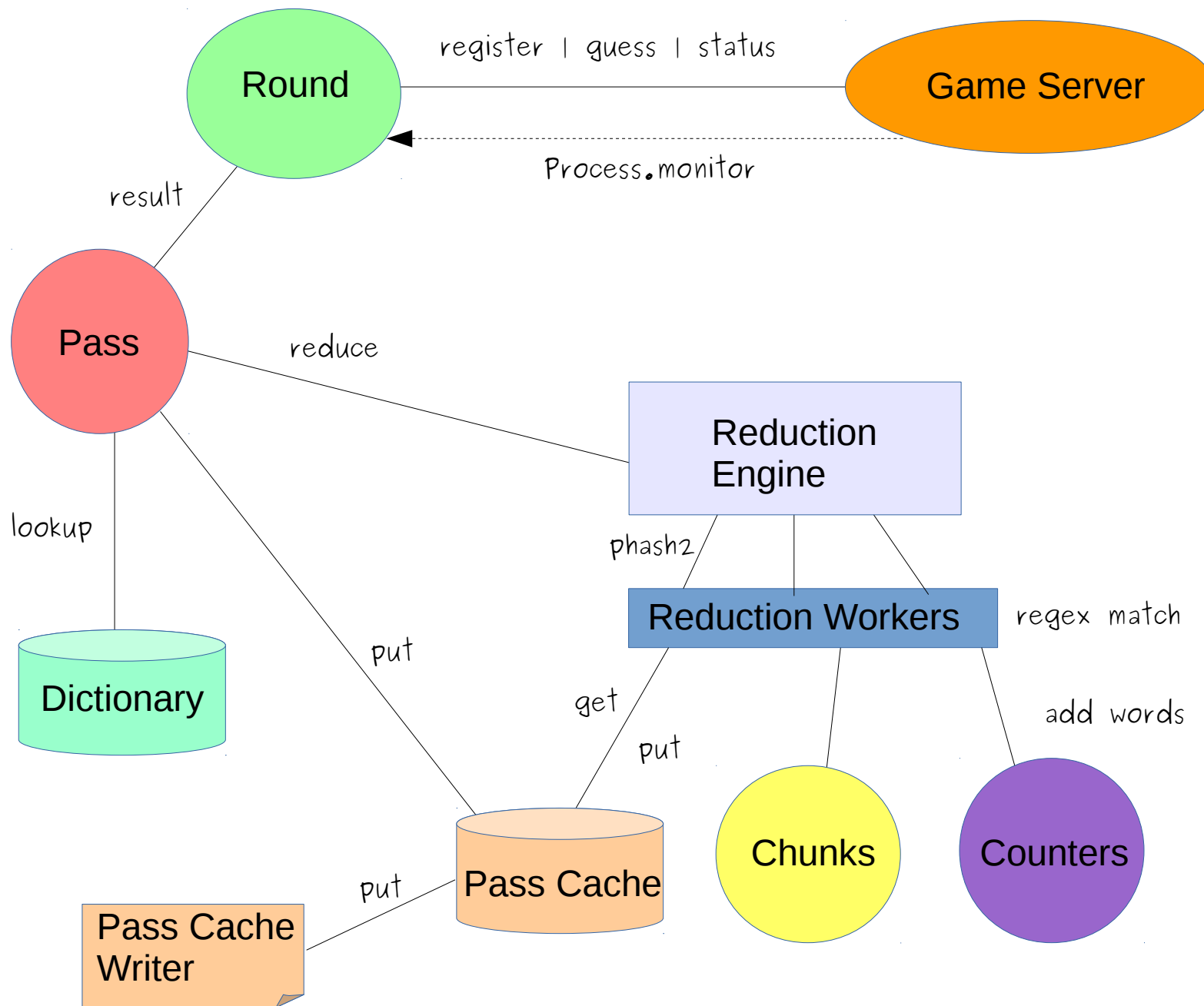
begin - start new game, if previous game is :abort and no games left → :exit

setup - setup round guess, if :human ask for letter input

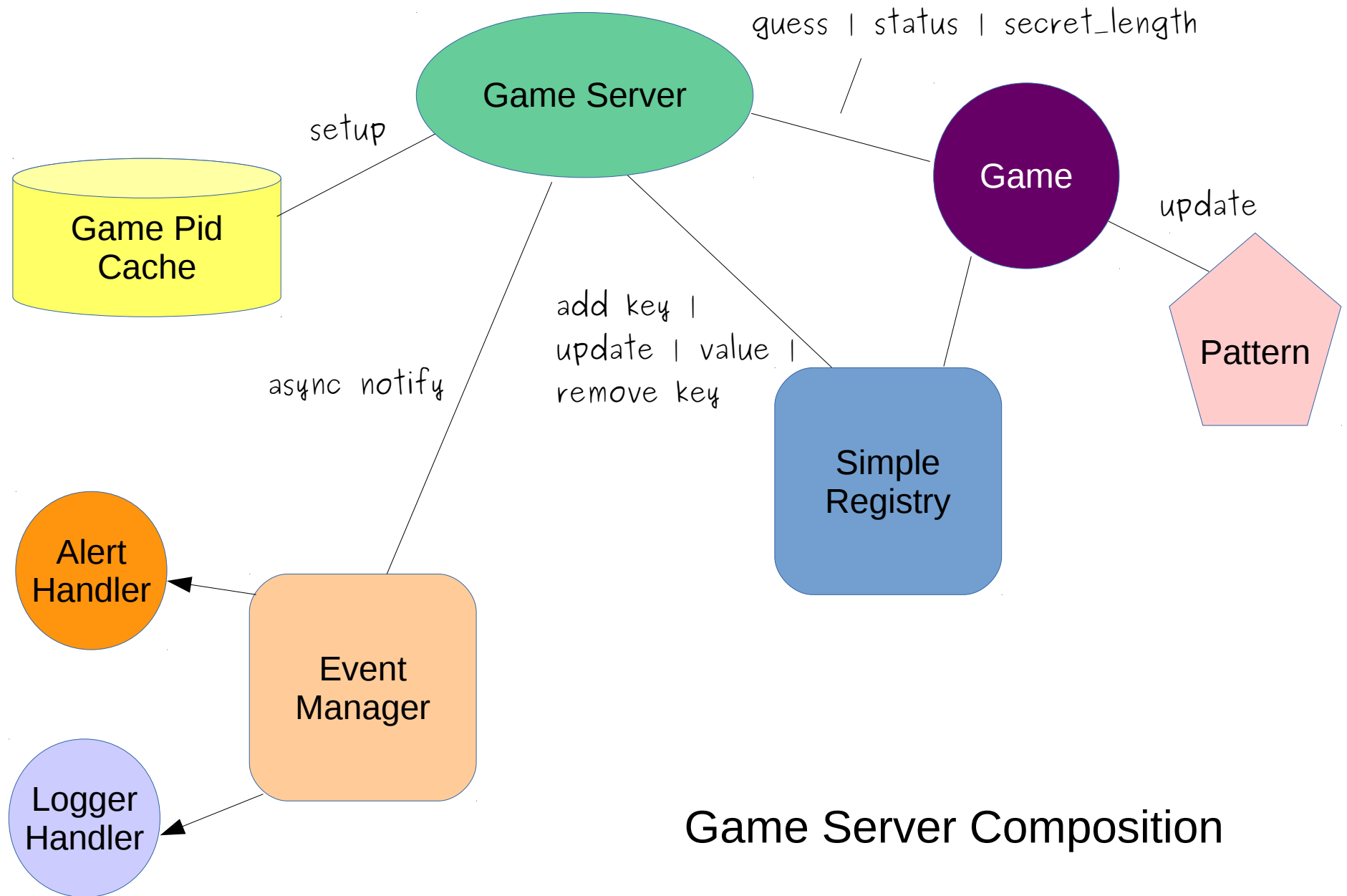
action - Perform guess, if more rounds left go back to :setup else :transit

transit - Single game over, start new game or exit if all games finished

exit - Games over!



Player System Interaction



Game Server Composition

# Hangman Supervision Tree

