

A Pattern for Distributing Turn-Based Games

James E. Heliotis, Axel T. Schreiner

Computer Science, Rochester Institute of Technology

OOPSLA Killer Examples Workshop, 2006.

<http://www.cs.rit.edu/~ats/papers/netgame/>

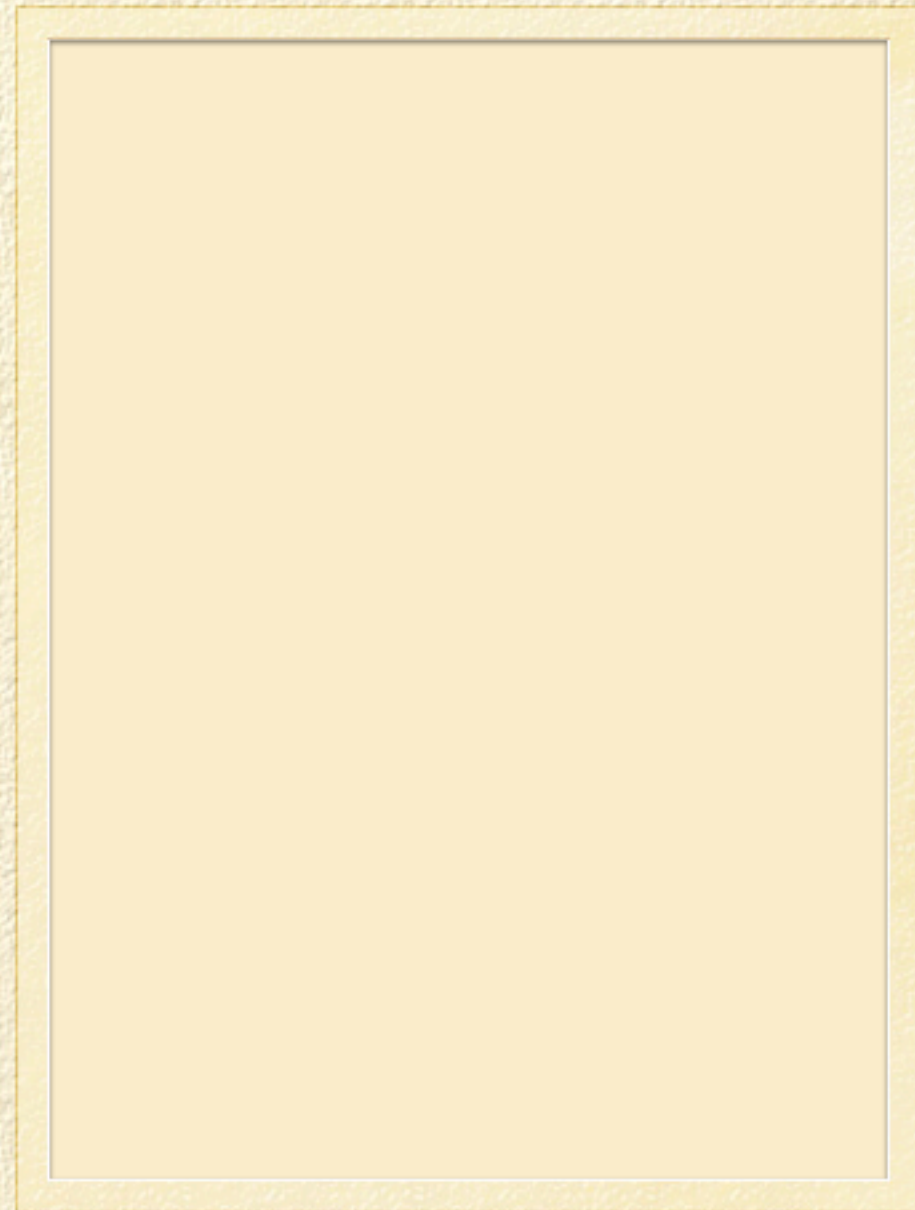
Background

- example from (advanced) Java and C# courses
- to discuss view/model communication by thread synchronization
- to promote non-trivial (binary) reuse
- to discuss client/server paradigm by example

Turn-taking games

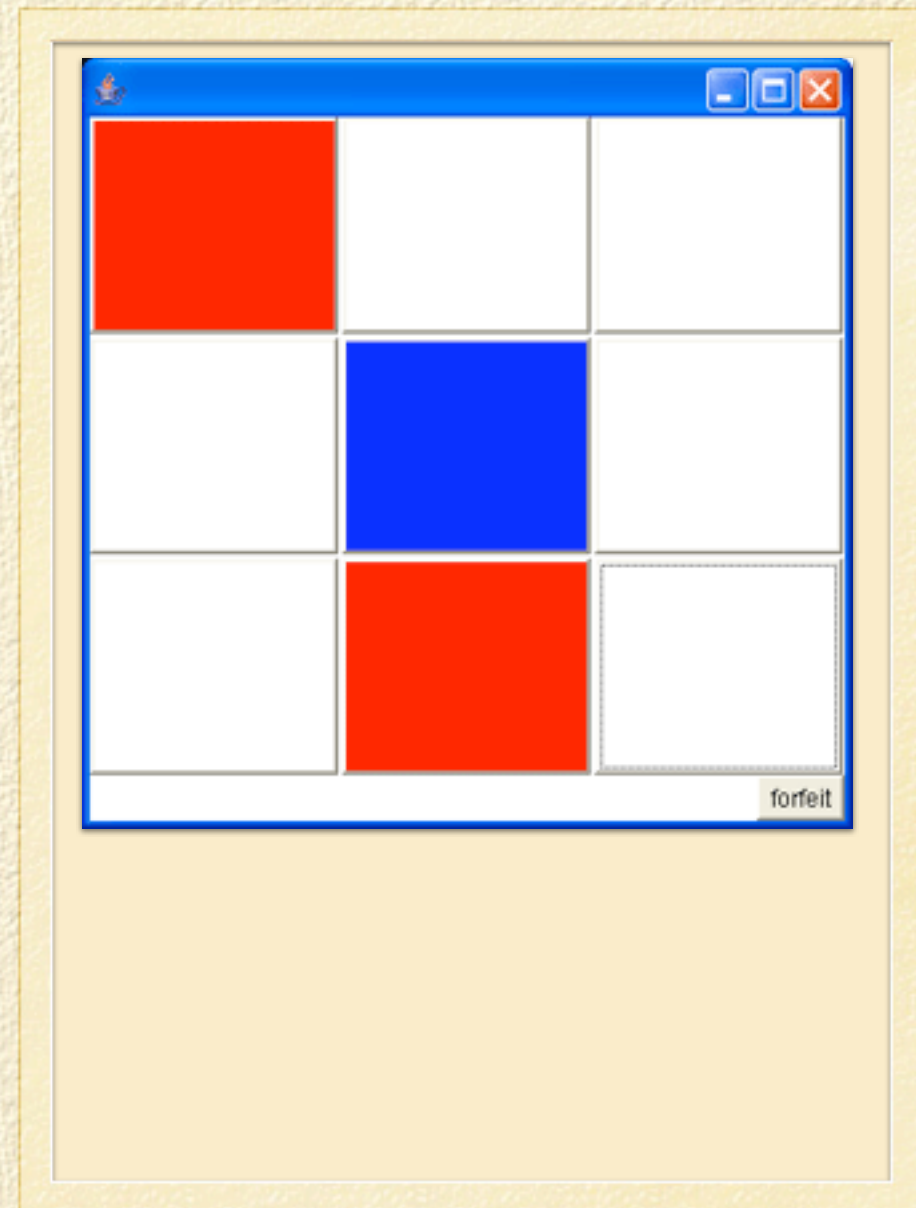
- two (or more?) players
- alternating (or concurrent?) moves
- global state (board)
- rule enforcement, e.g., win/loss, next up, ...

Examples



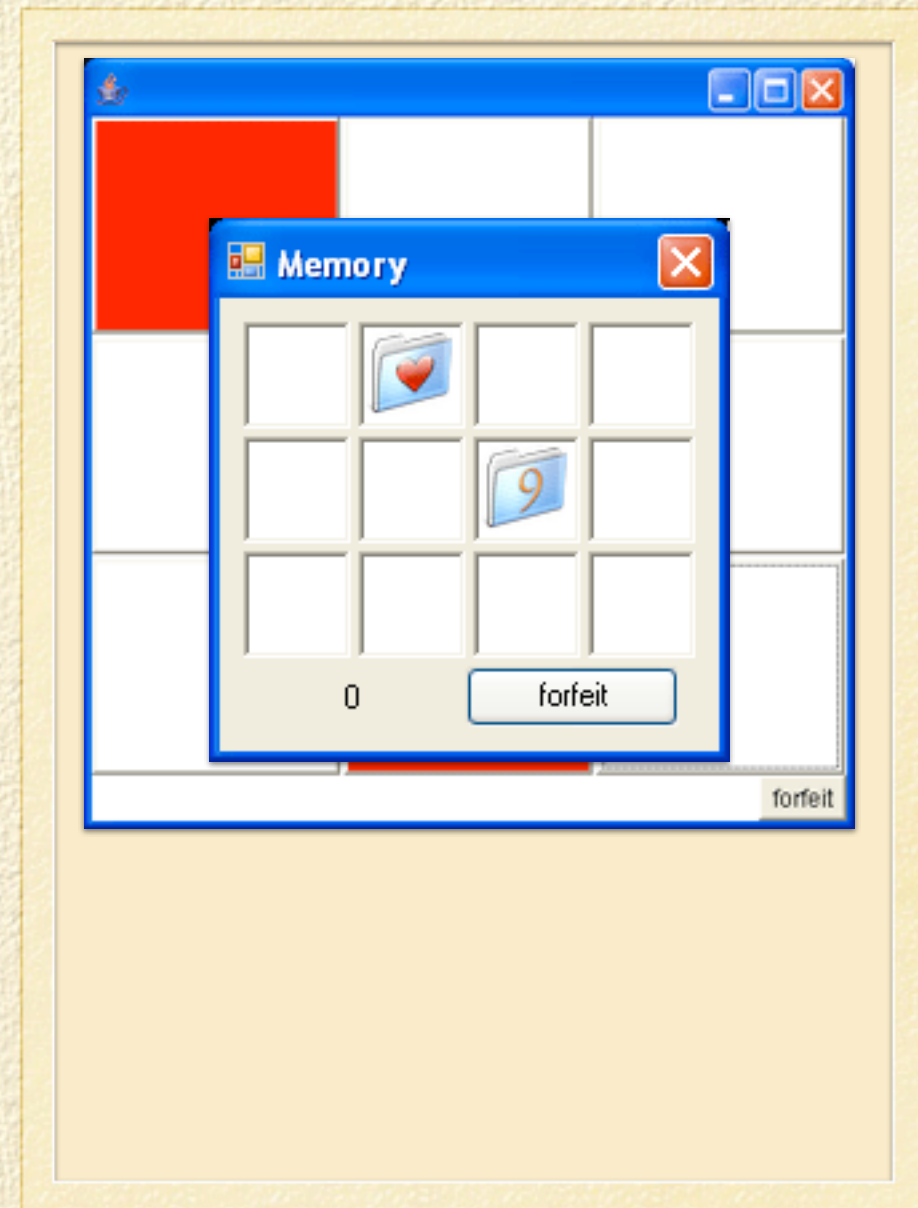
Examples

□ Tic-Tac-Toe



Examples

- Tic-Tac-Toe
- Memory



Examples

- Tic-Tac-Toe
- Memory
- Rock-Paper-Scissors



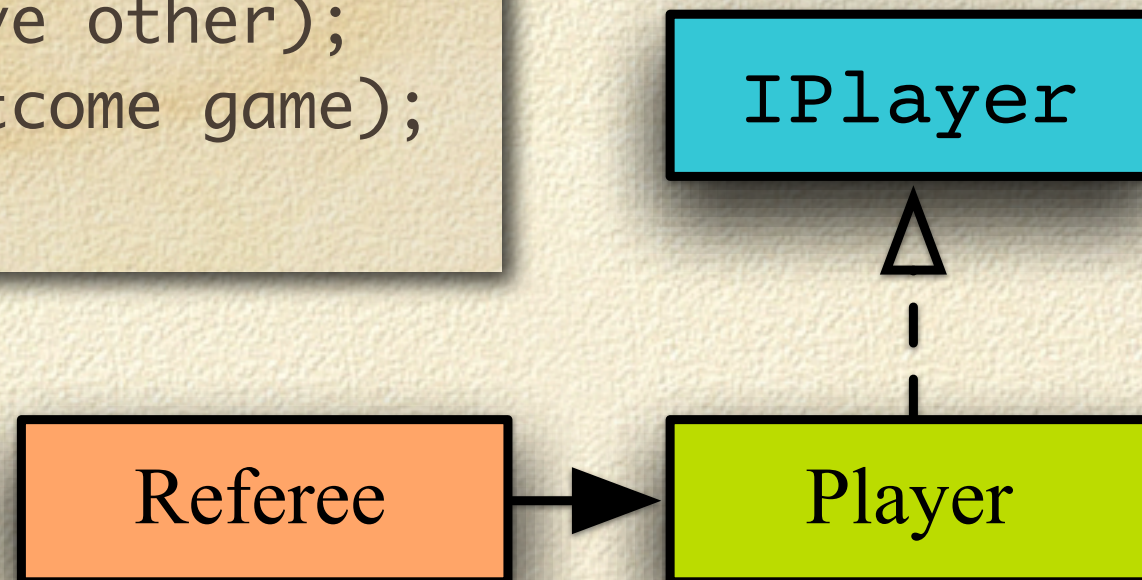
pre-OOP

```
main () {  
    player current = 0;  
    while (true) {  
        move m = get(current);  
        tell(m);  
        current = decide(m);  
    }  
}
```

- ❑ top-down
- ❑ passive players

OOP

```
interface IPlayer {  
    Move get ();  
    void tell (Move other);  
    void tell (Outcome game);  
}
```

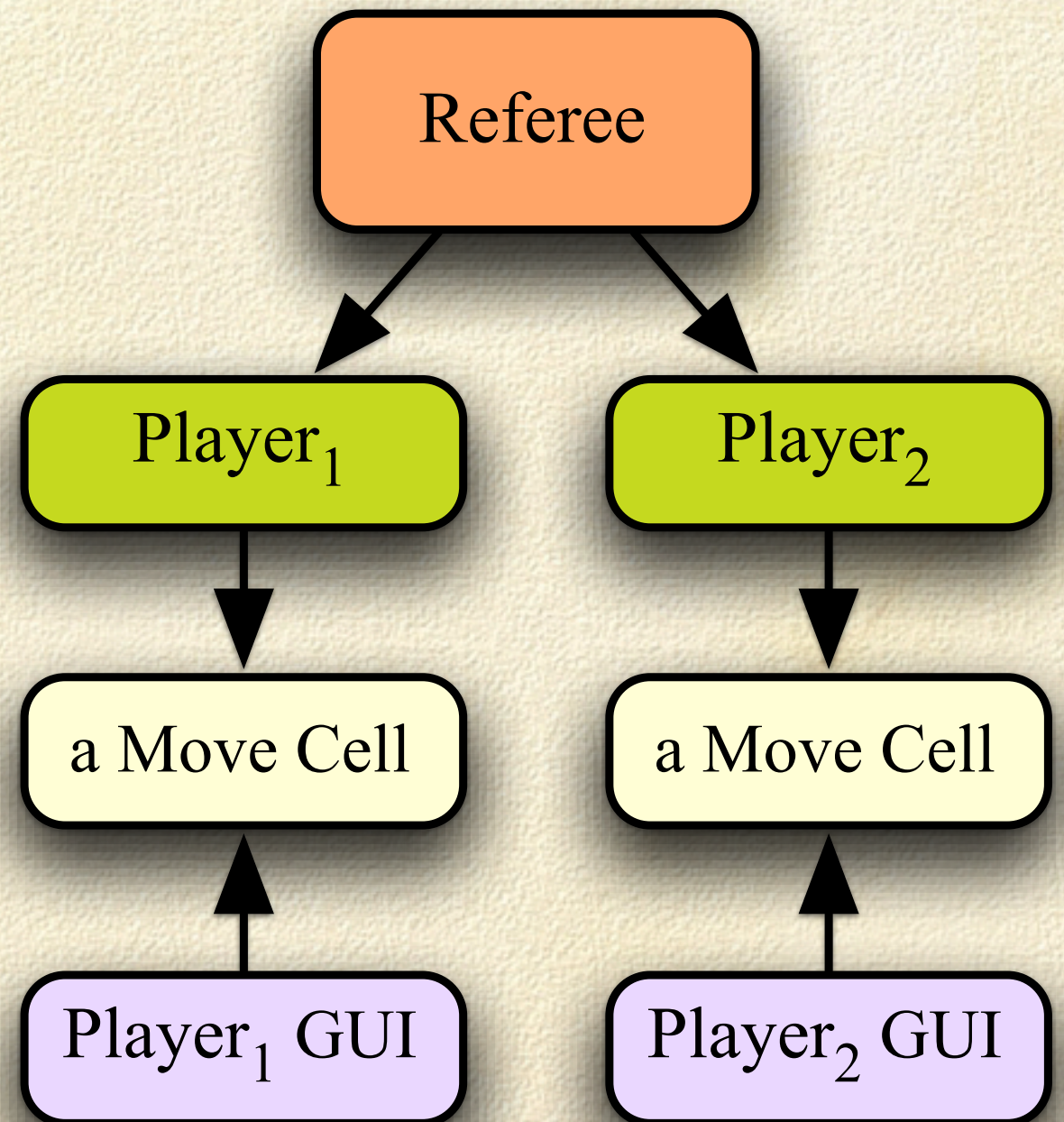


- stateless view
- active model (*not* observable)

Cell

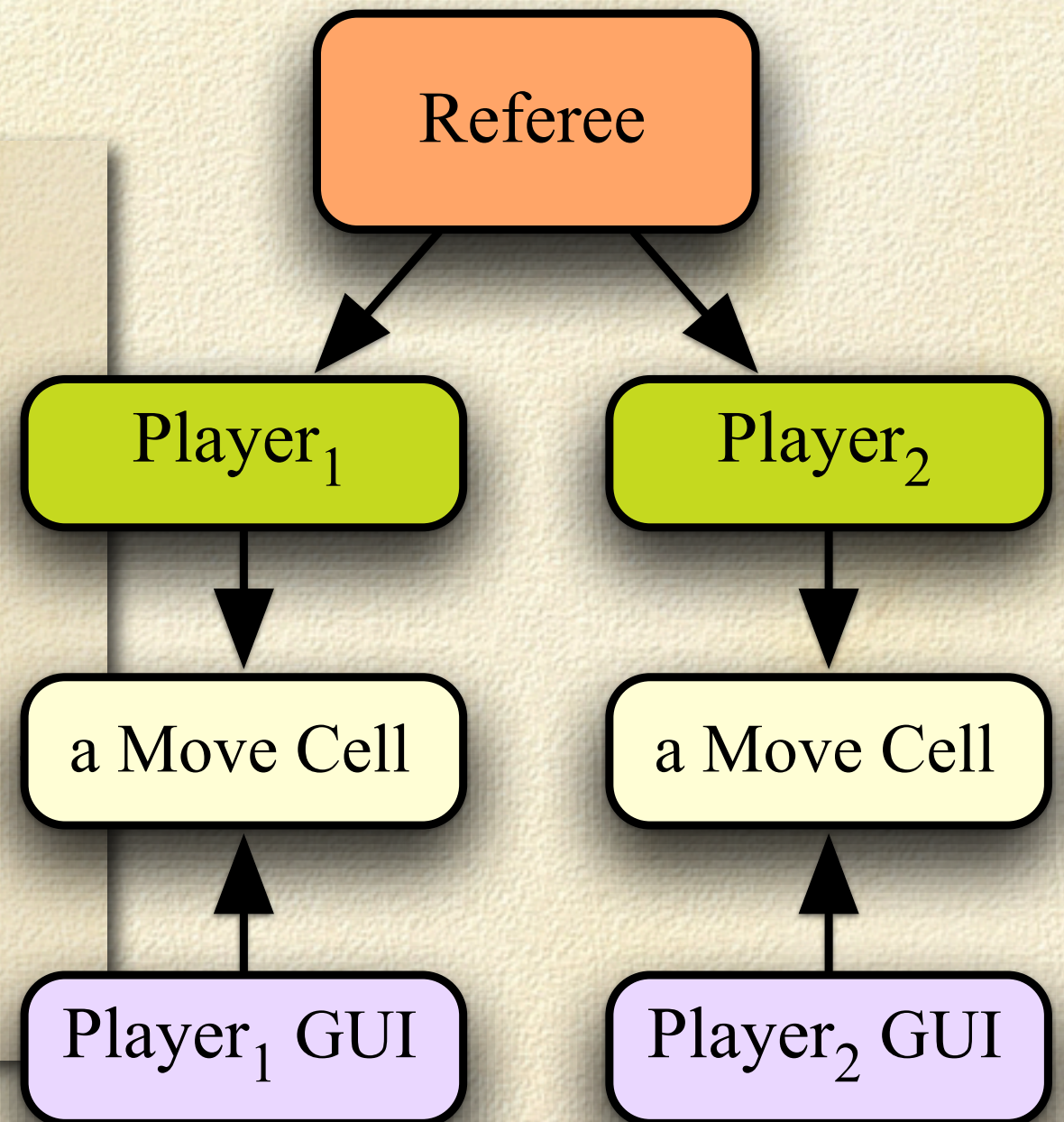
Cell

- model and view need to exchange a move

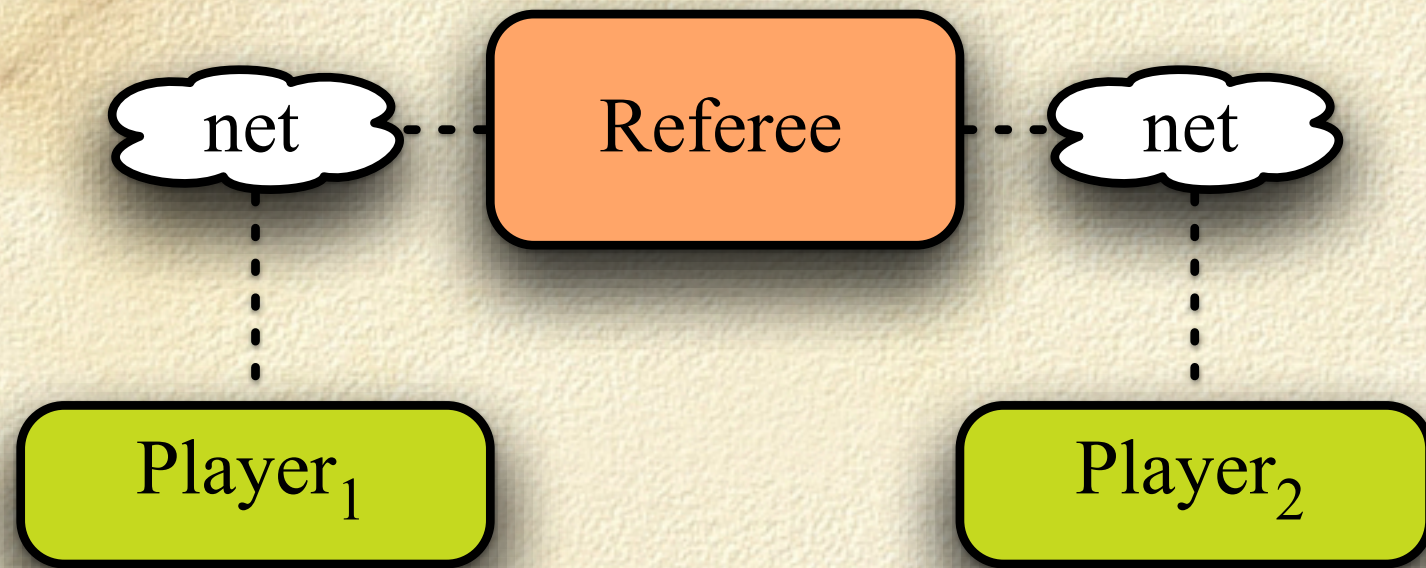


Cell

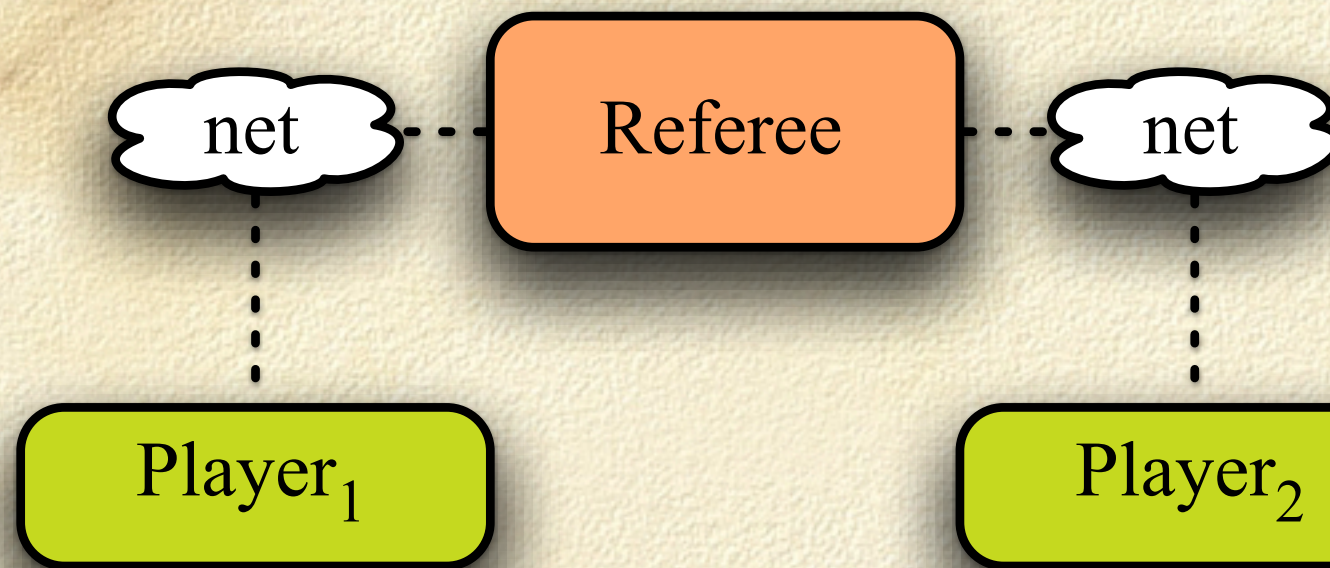
```
class Cell<T> {  
    T value; boolean full;  
    T synchronized get () {  
        while (!full) wait();  
        full = false; notifyAll();  
        return value;  
    }  
    void synchronized put (T value)  
        // ...  
}  
}
```



Distributed Game



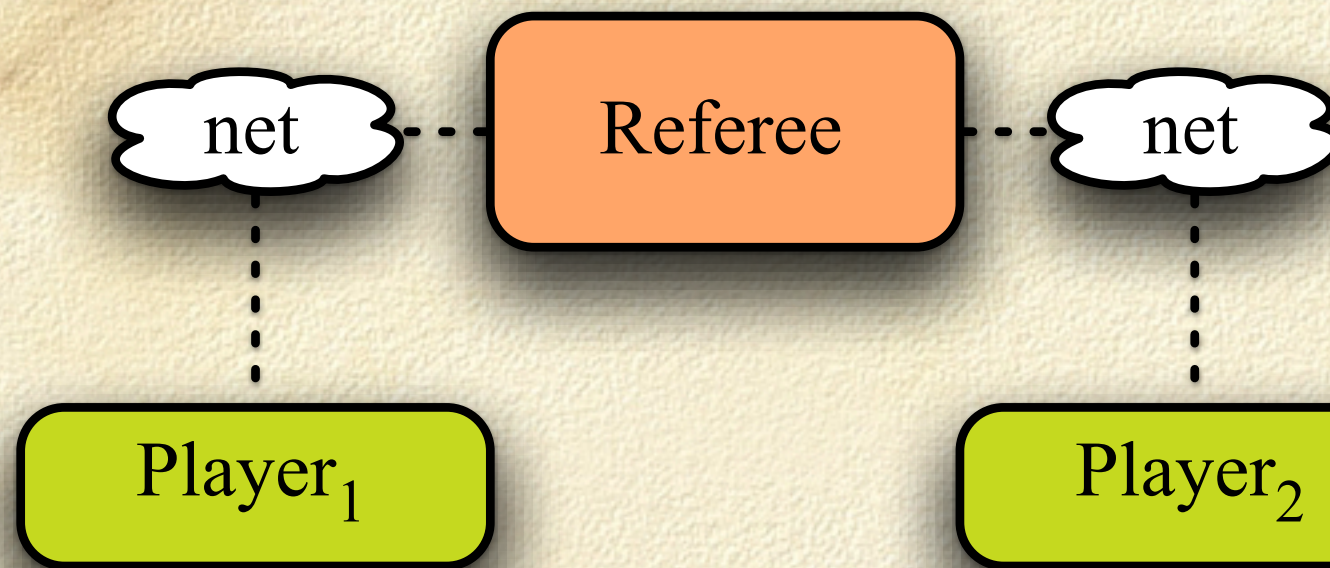
Distributed Game



```
interface IReferee {  
    void tell (Move mine);  
    Move getOther ();  
    Outcome get ();  
}
```

□ client calls on server

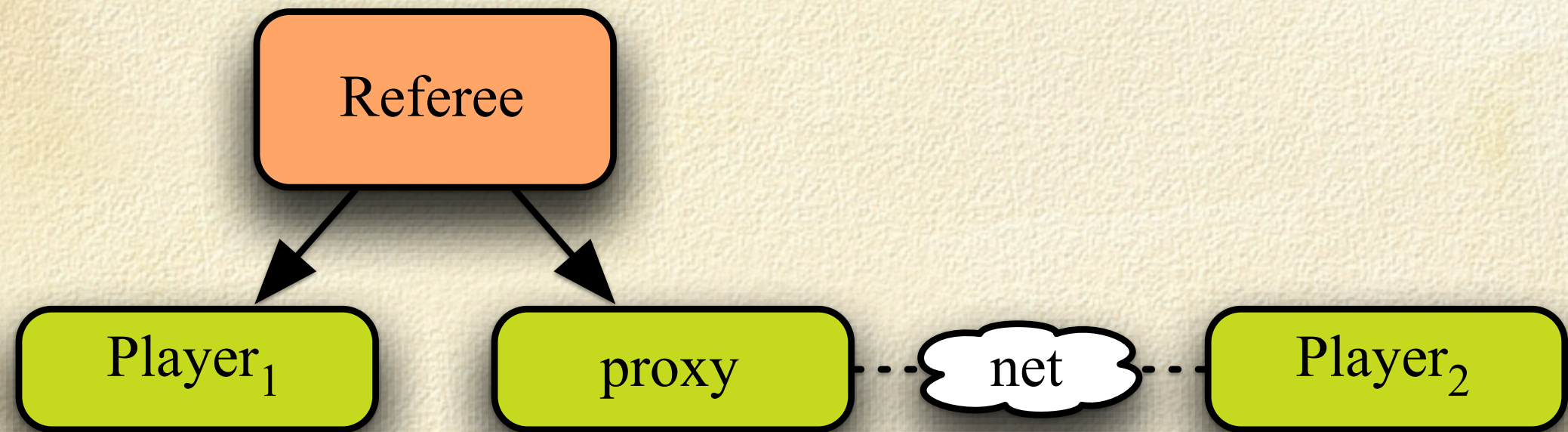
Distributed Game



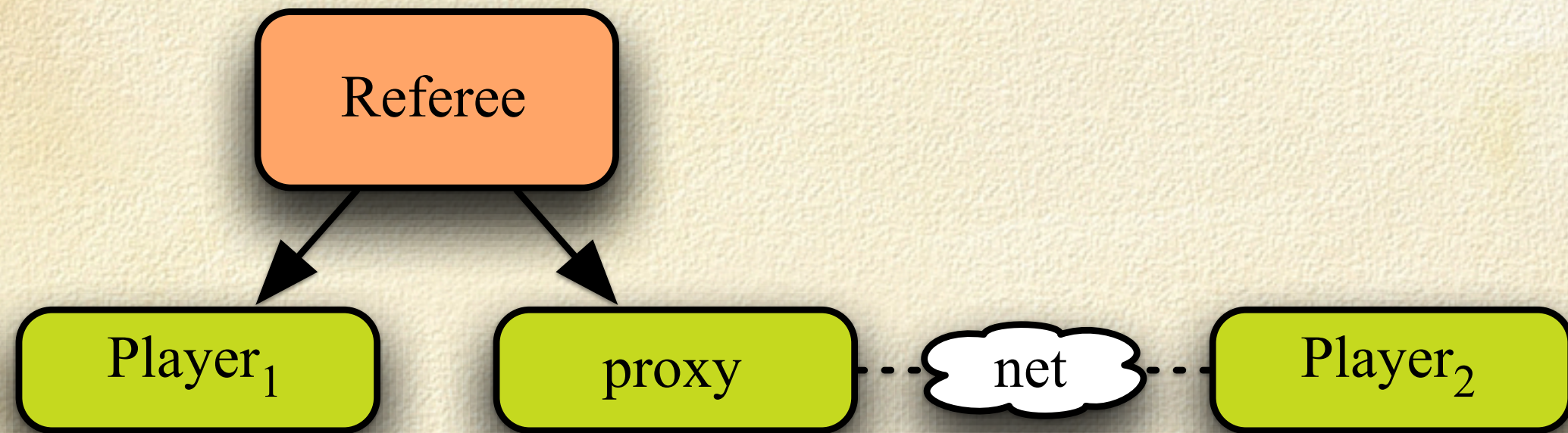
```
interface IReferee {  
    void tell (Move mine);  
    Move getOther ();  
    Outcome get ();  
}
```

- client calls on server
- *view would require state to sequence messages*

Proxy pattern

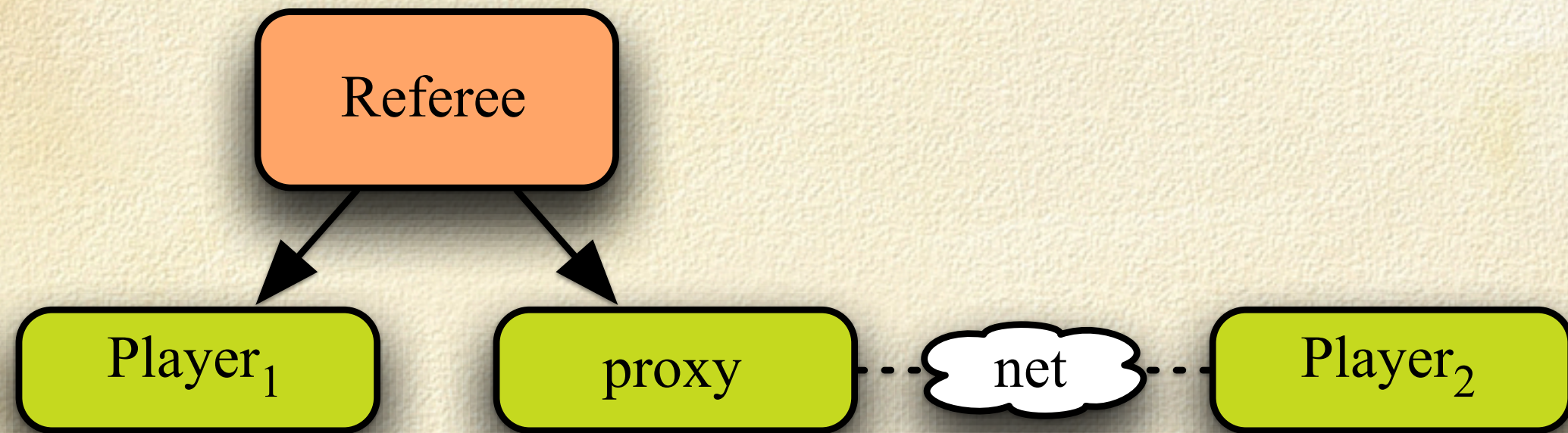


Proxy pattern



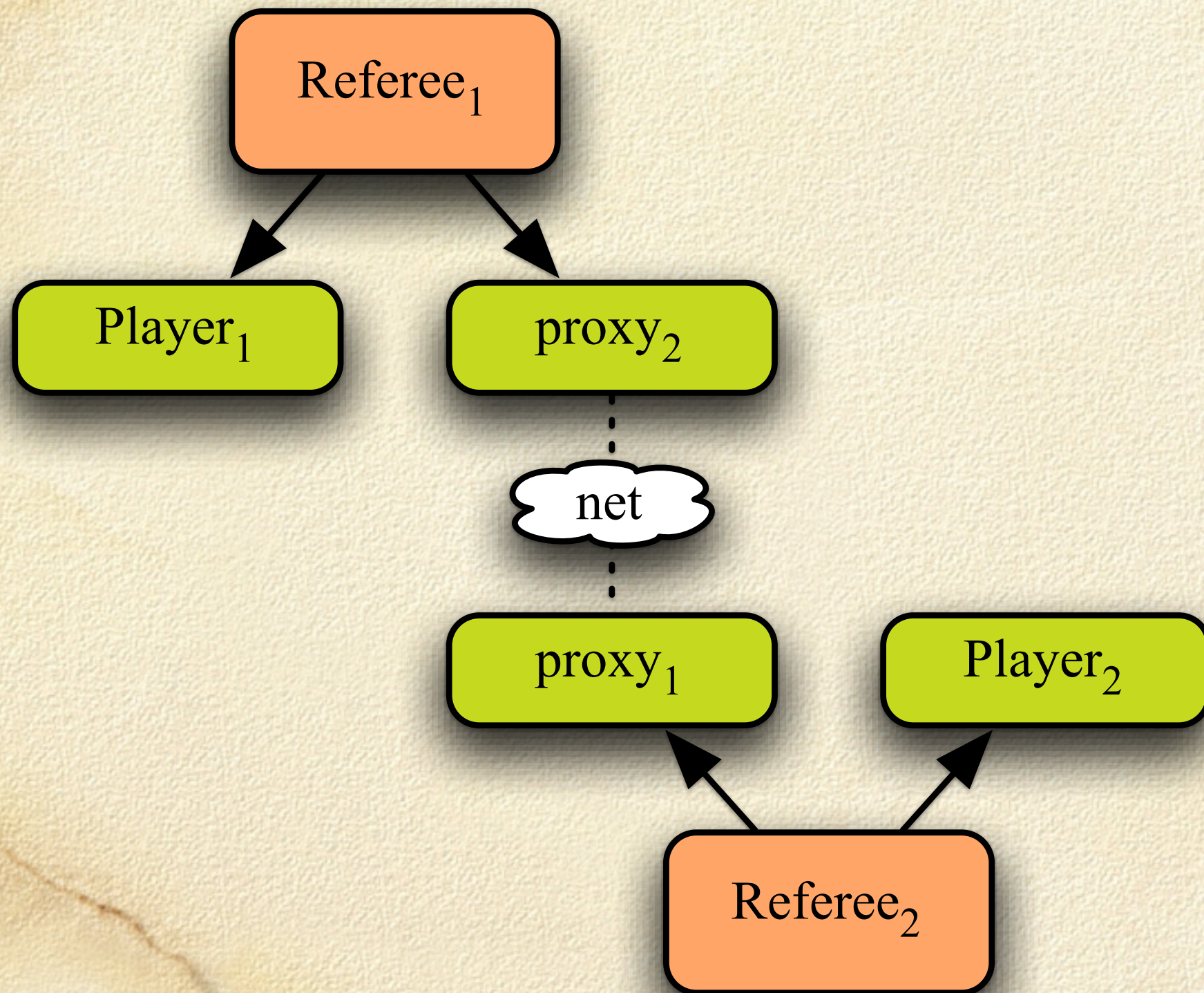
□ reinstates IPlayer

Proxy pattern

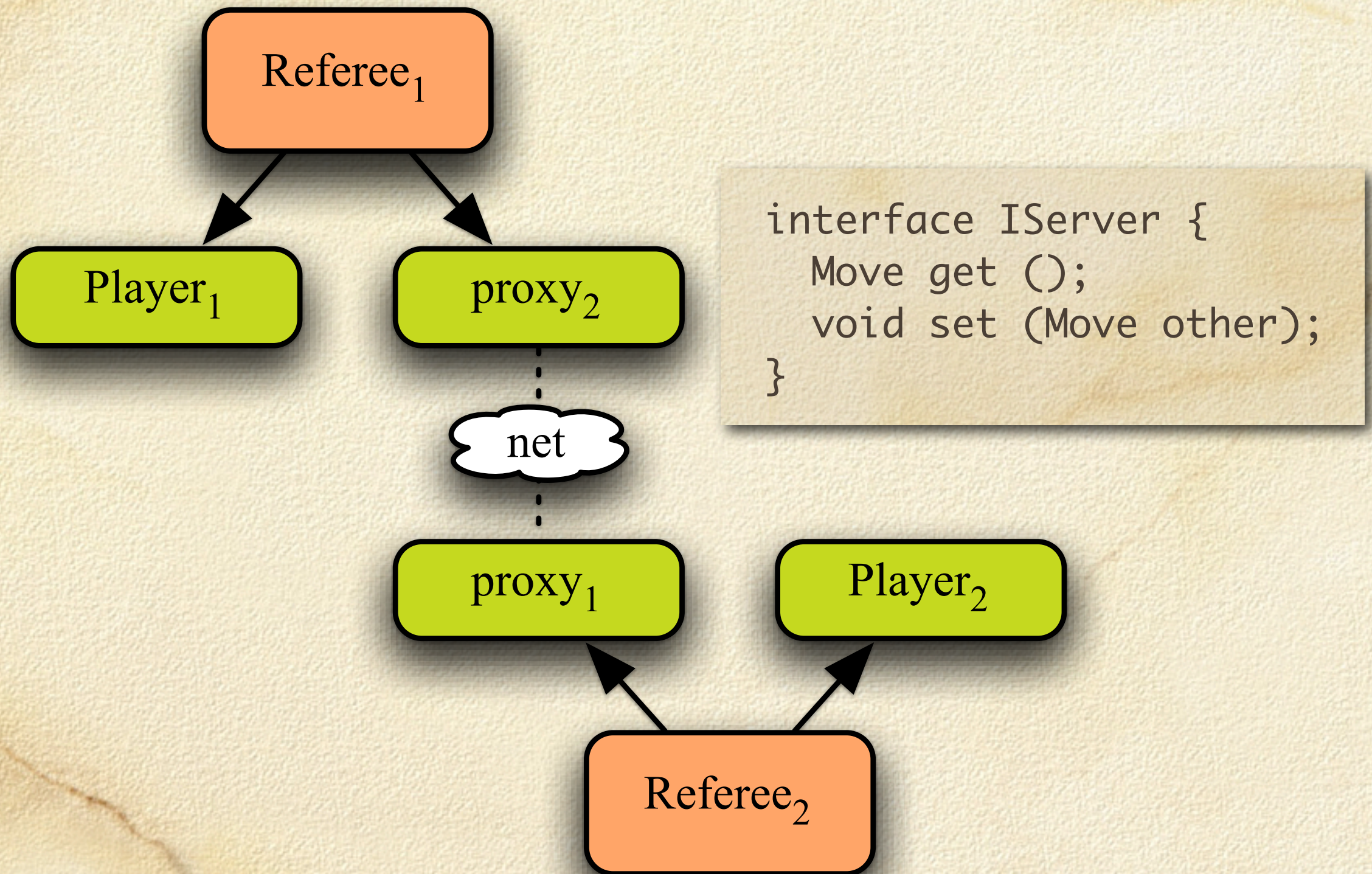


- ❑ reinstates IPlayer
- ❑ *would require net-aware second player (server)*

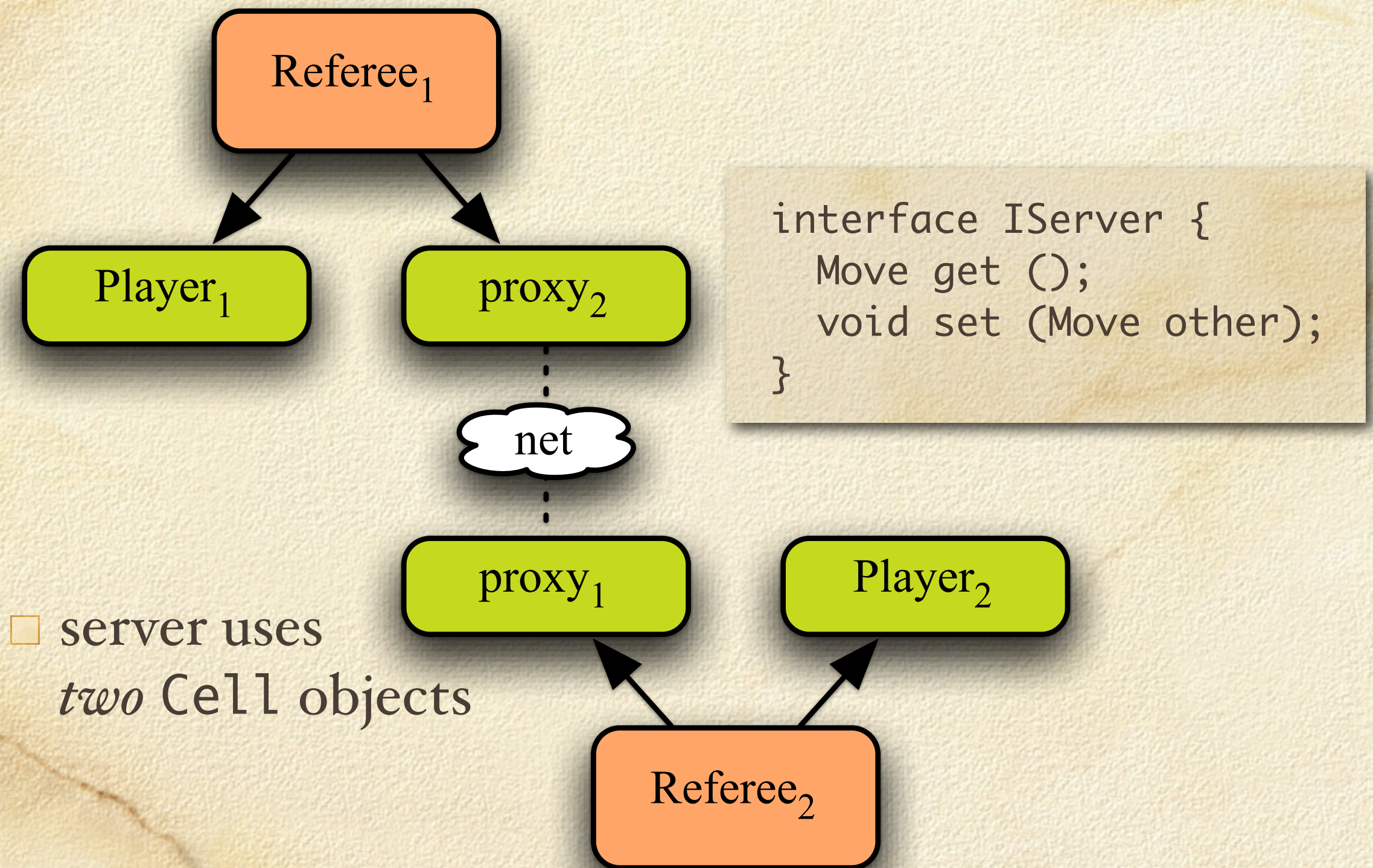
Symmetric proxy pattern



Symmetric proxy pattern

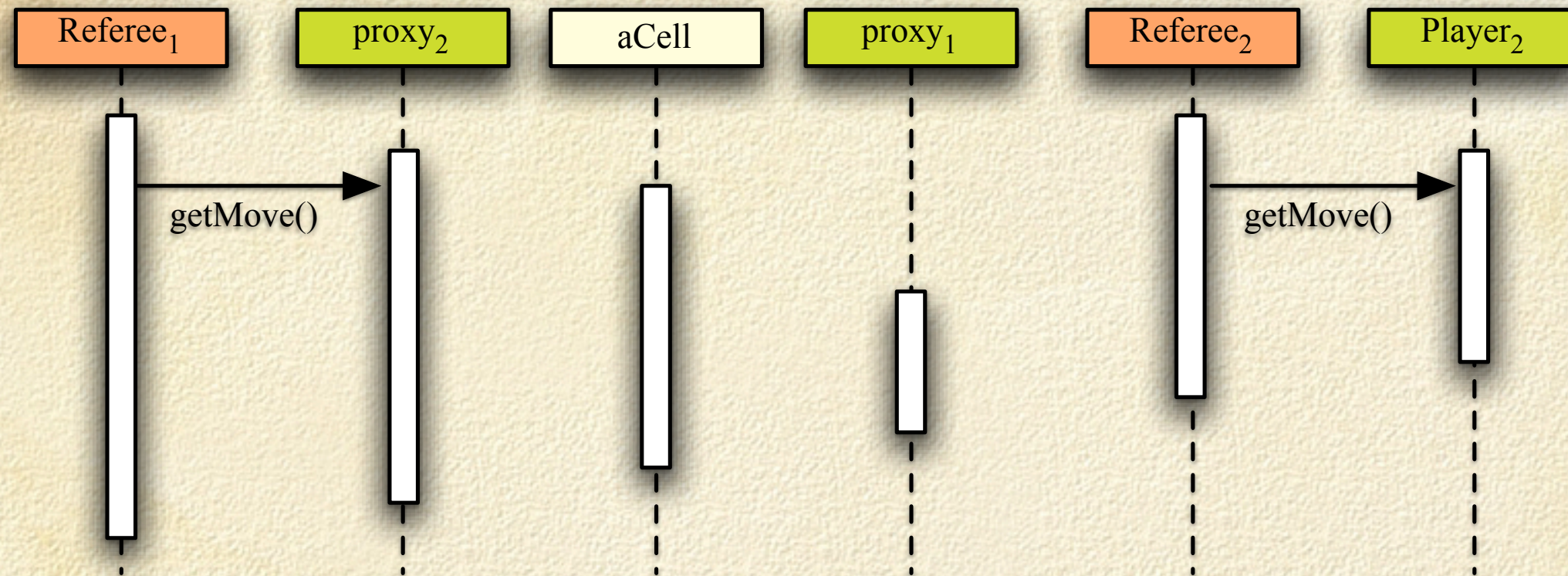


Symmetric proxy pattern



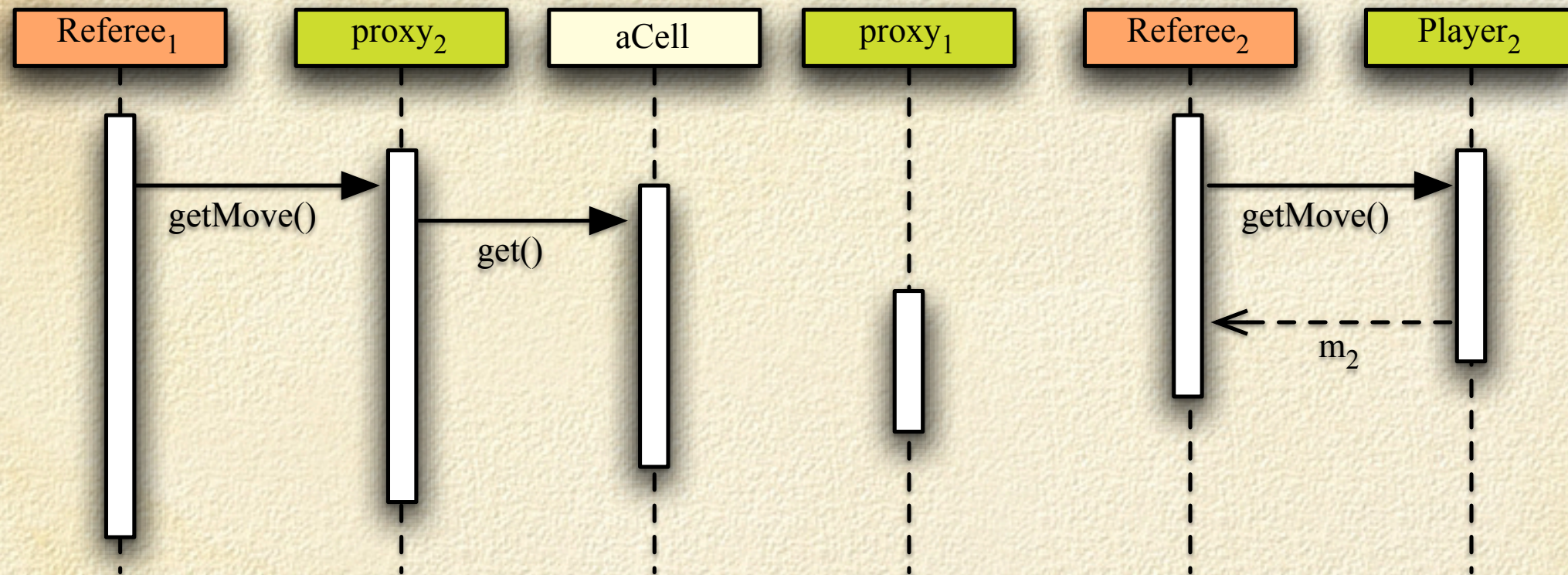
Alternate turn-taking

Alternate turn-taking



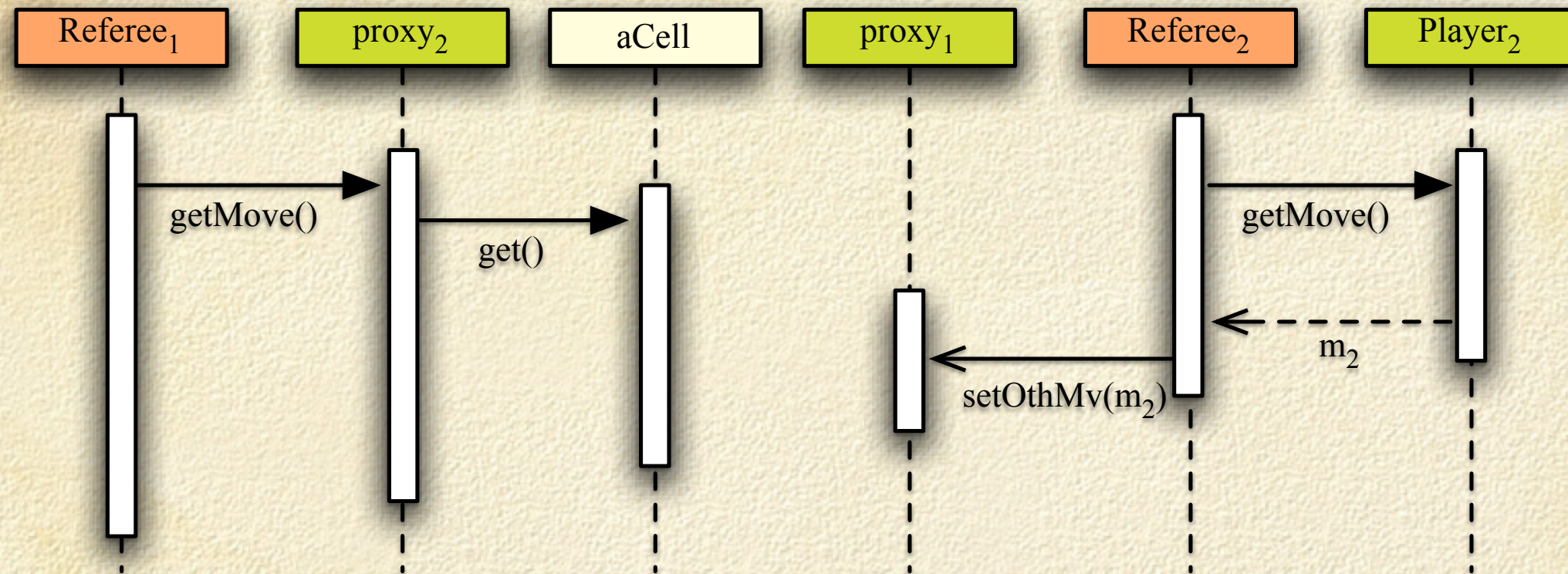
□ referees request move

Alternate turn-taking



- referees request move
- proxy waits for cell
- view provides move

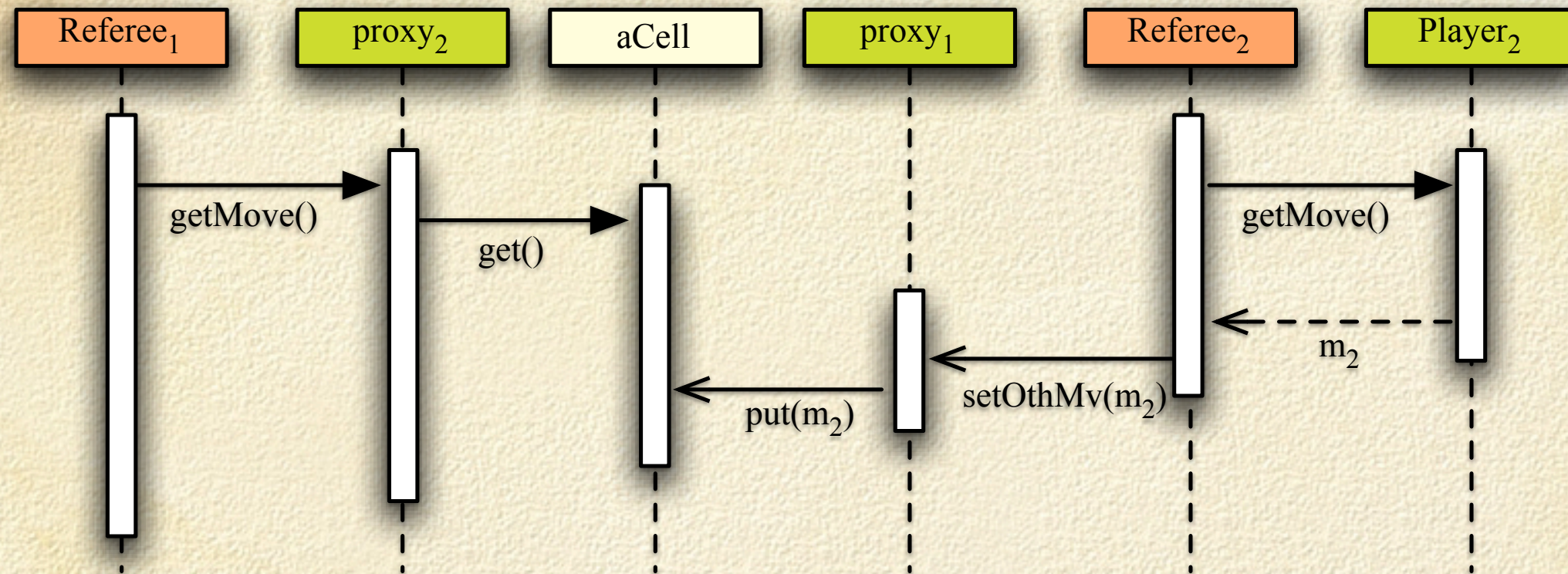
Alternate turn-taking



- referees request move
- proxy waits for cell

- view provides move
- referee tells move

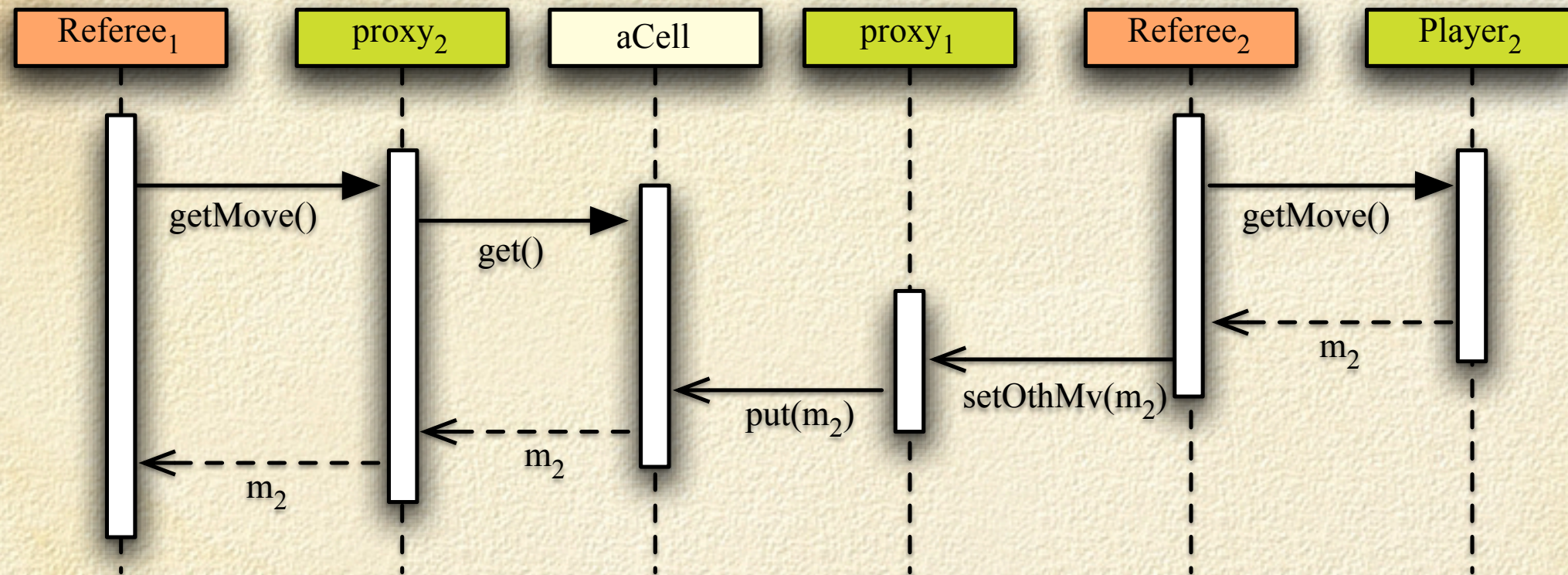
Alternate turn-taking



- referees request move
- proxy waits for cell

- view provides move
- referee tells move
- proxy forwards move

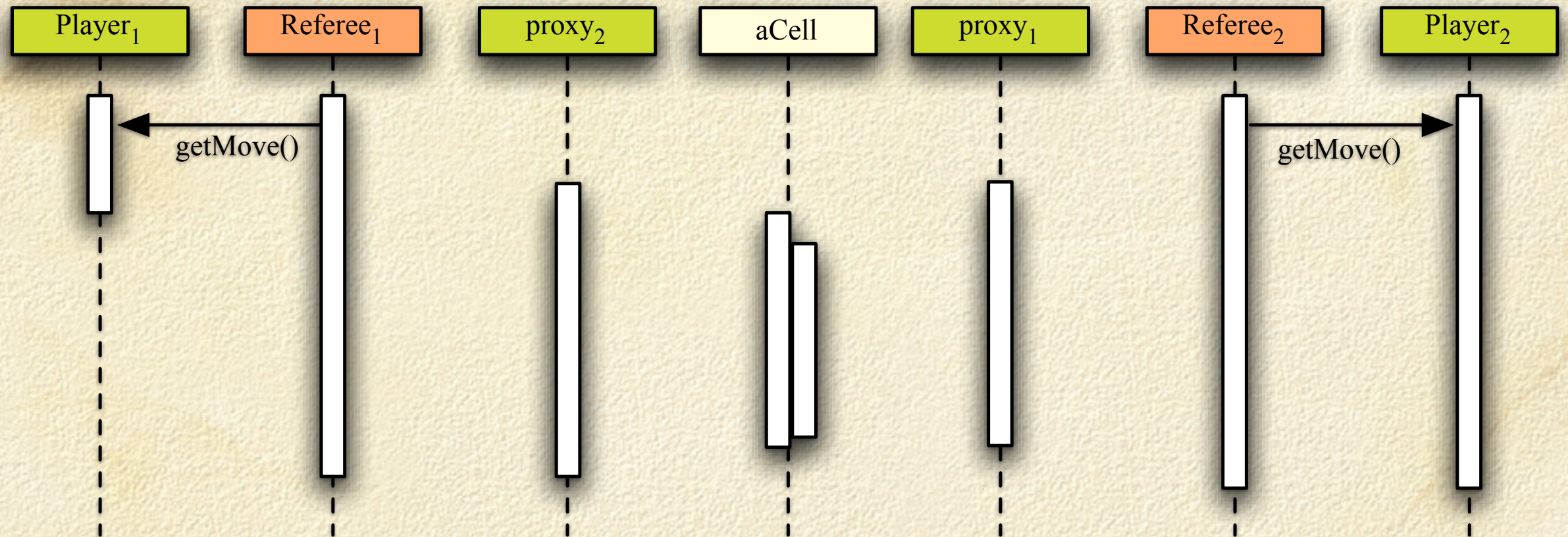
Alternate turn-taking



- referees request move
- proxy waits for cell
- referee receives move
- view provides move
- referee tells move
- proxy forwards move

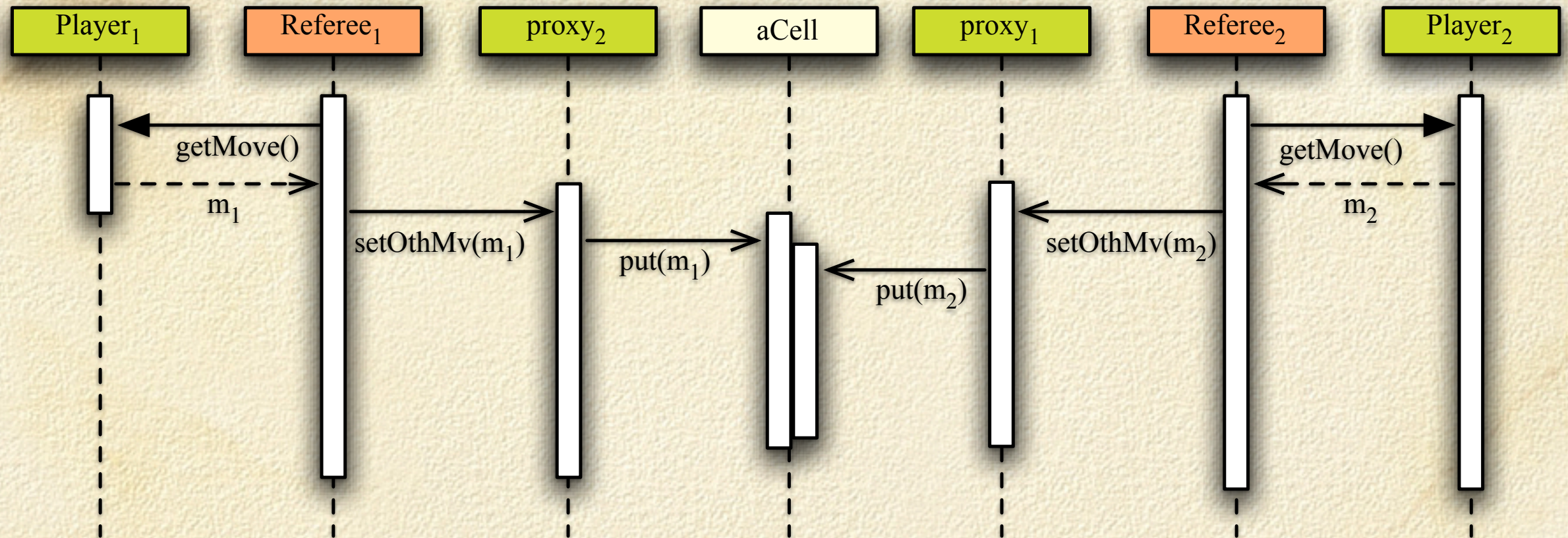
Concurrent turns

Concurrent turns



□ referees request moves

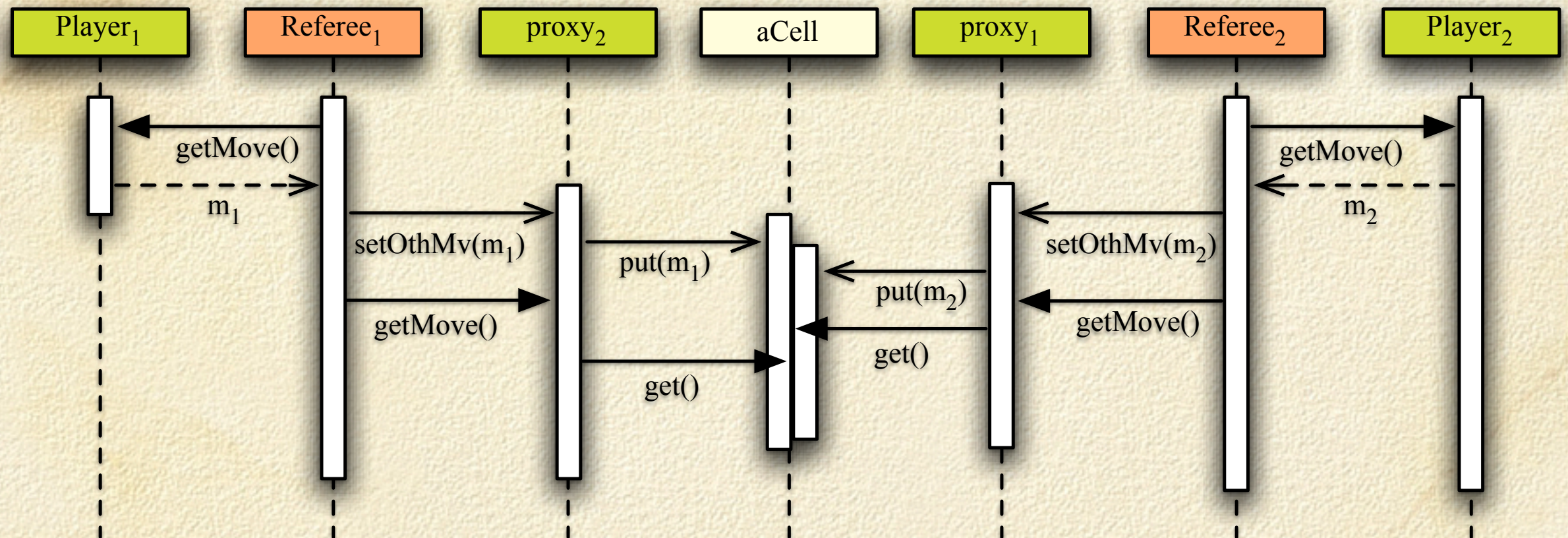
Concurrent turns



- referees request moves
- referees tell moves

- views provide moves
- proxies forward moves

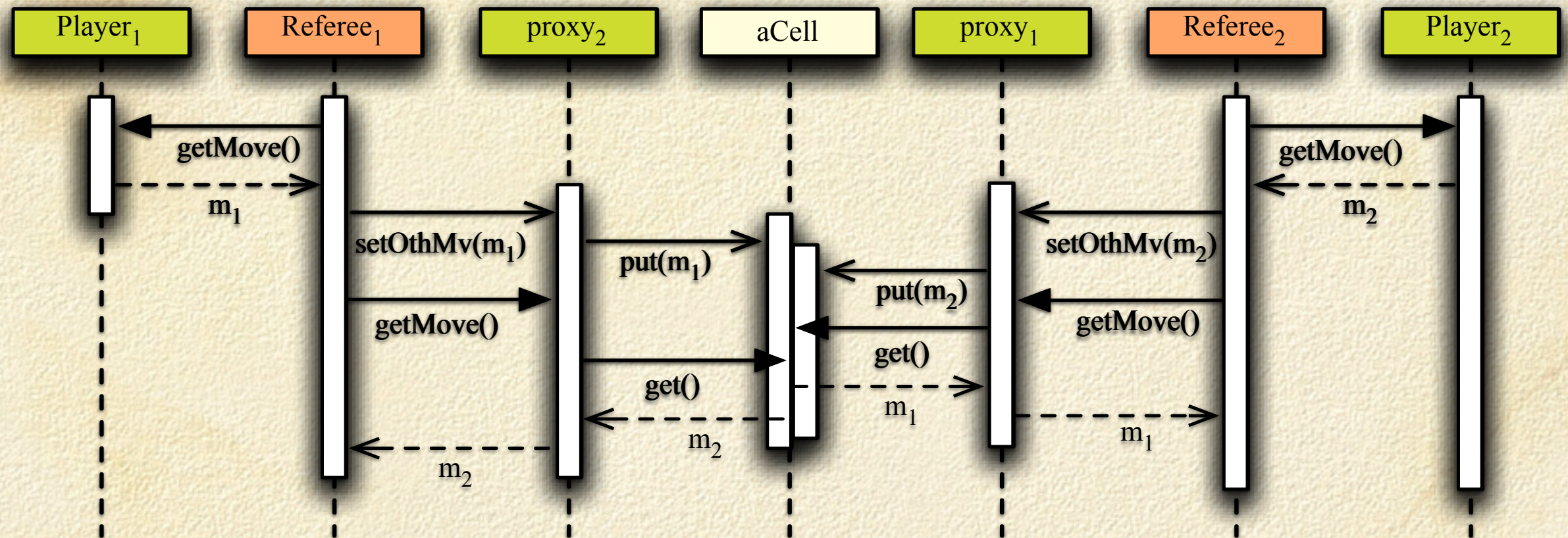
Concurrent turns



- referees request moves
- referees tell moves
- referees request moves

- views provide moves
- proxies forward moves

Concurrent turns



- referees request moves
- referees tell moves
- referees request moves

- views provide moves
- proxies forward moves
- referees receive moves

Links

<http://www.cs.rit.edu/~ats/papers/netgame/links.html>

Demo4			
Java TCP	all in one	papers/netgame/Demo4.java	•
Memory			
C# SOAP HTTP	Model	cs-2006-1/3/solved.html	•
	View	cs-2006-1/4/solved.html	•
	Proxy	cs-2006-1/5/solved.html	•
	observable	cs-2004-1/problems/4/	•
Java TCP	Model View	java-2005-2/8/solved.html	•
	Proxy	java-2005-2/9/solved.html	•
Rock-Paper-Scissors			
C# SOAP HTTP	Model View Proxy	cs-2006-1/6/solved.html	•
	outcomes	cs-2005-1/3/solved.html	•
Tic-Tac-Toe			
C# SOAP HTTP	Model/FSA	cs-2005-1/5/solved.html	•
	Local	cs-2005-1/6/solved.html	•
	Proxy	cs-2005-1/7/solved.html	•
all links relative to http://www.cs.rit.edu/~ats/			