

DISTRIBUTED PROGRAMMING IN ERLANG

Curt Clifton
Rose-Hulman Institute of Technology

SVN Update *ErlangInClass*,
Open *kvs/kvs.erl*

FINAL PRESENTATIONS

- Starting in class on Friday
- 25 minutes for each team
- See week 10 milestone for detailed description

SCHEDULING PRESENTATIONS

	Friday 9th Week	Monday 10th Week	Tuesday 10th Week	Thursday 10th Week
1st half of class				
2nd half of class				

Breakfast
Club

Groove
Monitors

Team
Awejome

The Peculiar
Institution

Team Big
Mouth

Helpdesk
Fish

1991 in
Norway

A large, textured red rectangular area with a slightly irregular, hand-painted appearance, set against a light gray background. The red area has a fine, grainy texture and a subtle gradient, with the top edge appearing slightly more saturated than the bottom.

WHY WRITE DISTRIBUTED APPLICATIONS?

TWO MAIN MODELS IN ERLANG

- Distributed Erlang
- Socket-based communication

EXAMPLE

- Key-Value server on one machine

COMMAND-LINE ERLANG ON WINDOWS

- Find fully qualified path to **erl.exe**, something like:
 - **C:\Program Files\erl5.7.3\bin**
- Edit your PATH environment variable to add that:
 - Right-click **My Computer**, then choose **Properties** → **Environment Variables**
 - Separate entries in PATH with semicolons
- Use **Start** → **Run** → **cmd**, then enter **erl** at prompt

EXAMPLE CONTINUED

- Client and Server on two nodes of same machine
- Client and Server on different machines

DANGER WILL ROBINSON!

- Machines with shared cookies can send **ANY** command to each other



BIFS FOR DISTRIBUTED PROGRAMMING

- `spawn(Node, Mod, Func, ArgList)`
- `spawn_link(Node, Mod, Func, ArgList)`
- `disconnect_node(Node)`
- `monitor_node(Node, Flag)`
- `node()`, `nodes()`
- `node(Arg)`

EXAMPLE

- Adding remote spawning to key-value server