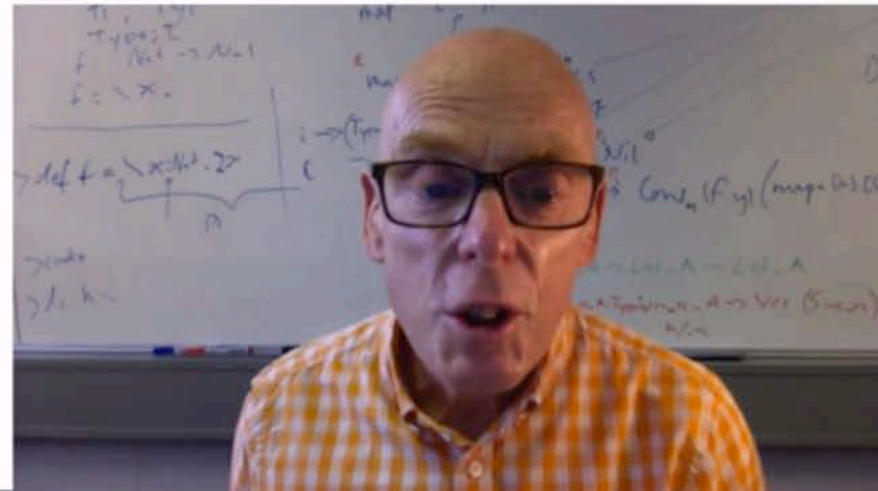

University of
Kent

University of Kent

Distributed Erlang



University of Kent

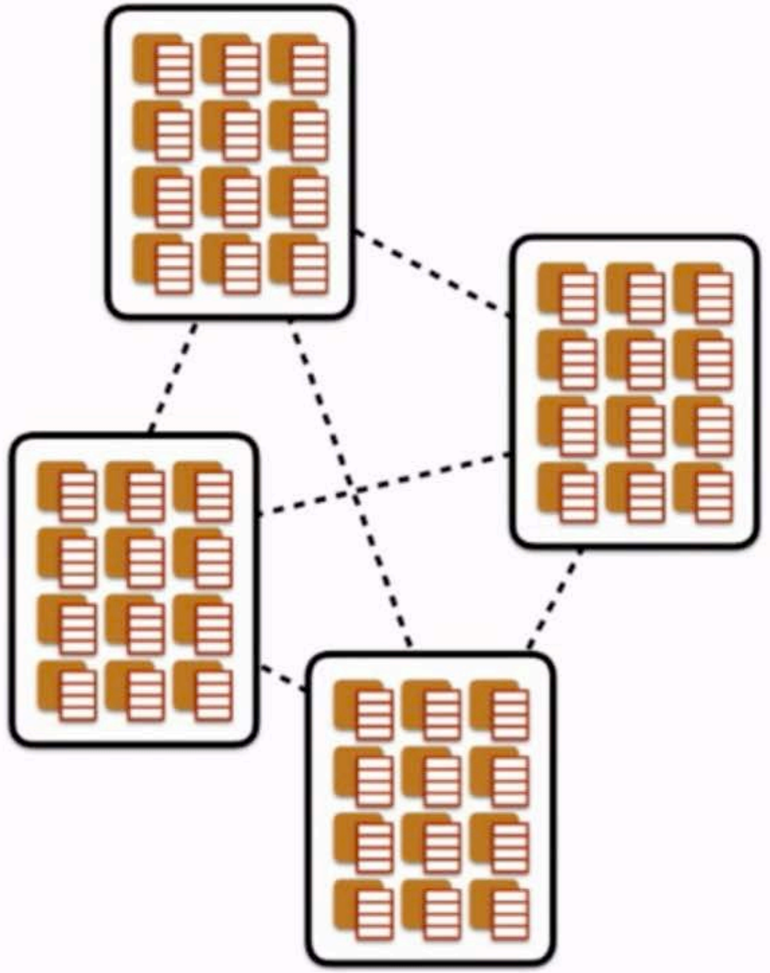
Distributed Erlang

Multiple Erlang instances …
…typically running on different machines or *hosts*.

Fully connected by default.

Communication by message passing to Pids and named processes.

Security through a shared secret (*"cookie"*).

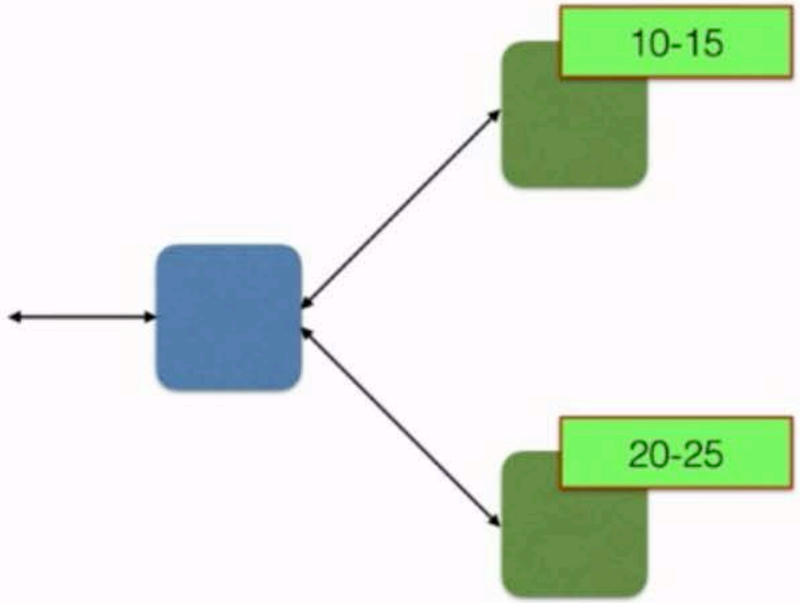


University of Kent

Scaling up the frequency server

Front-end routes requests to servers 1 (10-15) and 2 (20-25) … and replies to the client.

Front-end can track which servers have available frequencies … and how many each.



University of Kent

How are nodes named?

A node is started with the *short name* `ant` like this:

```
erl -sname ant -setcookie bee
```

and it can have the cookie value `bee` set at the same time.

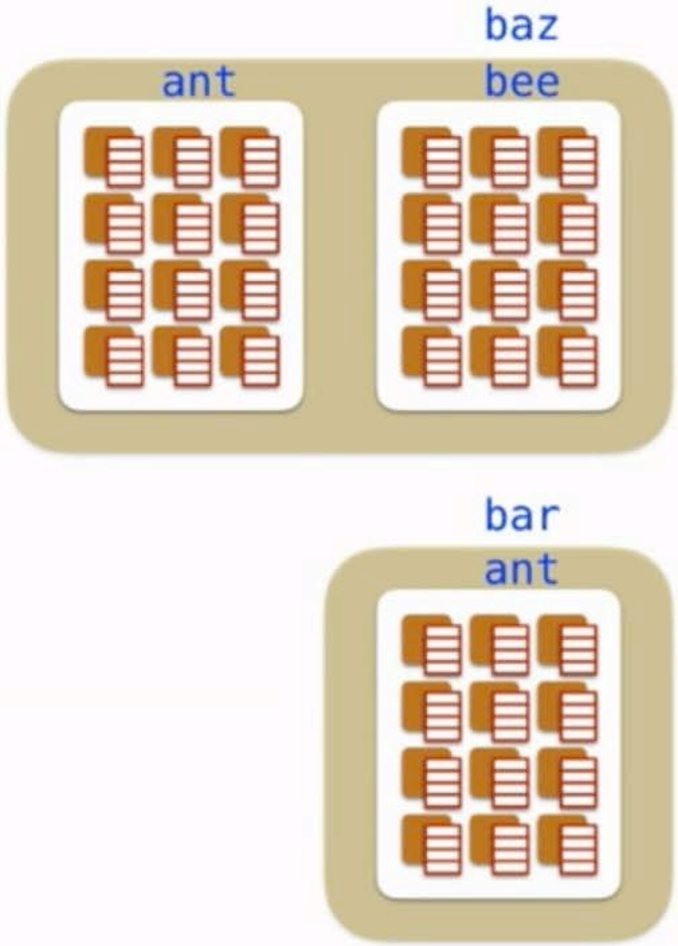
Short-named nodes communicate on the local network.

Long named nodes – e.g. `ant@192.168.11.121` – communicate globally, and can use DNS to resolve e.g. `ant@cs.kent.ac.uk`

Long names and short names can't be mixed in one system.

University of Kent

How are nodes identified?



A node started with the *short name* `ant` on a host `baz` will be identified by the atom `'ant@baz'`.

The same name can be used on different hosts, but only once on a single host … try starting two!

How to find out your host name? It's in the prompt:

```
% erl -sname foo
Erlang/OTP 18 [erts-7.3] … …(foo@edu72AD)1>
```

University of Kent

Communication and spawning

To communicate with a process `Pid` on another node:

```
Pid ! Message
```

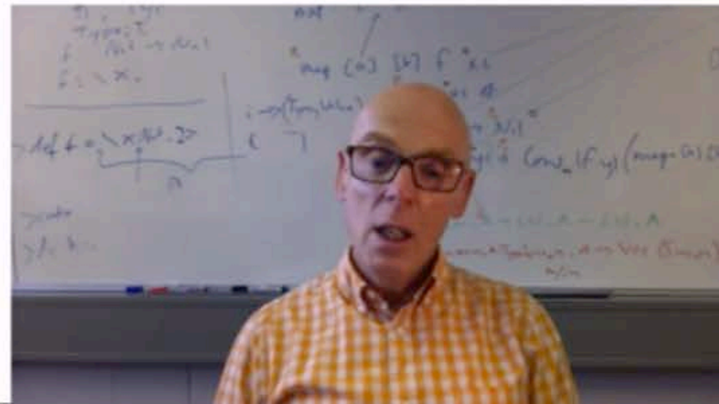
If a process is **named** `elf` on the node `ant` on host `baz` use:

```
{elf, 'ant@baz'} ! Message
```

To spawn a process on another node:

```
Pid = spawn('ant@baz',M,F,A)
```

To name a process on another node, need for the name to be registered at that node.



University of Kent

Communication and spawning

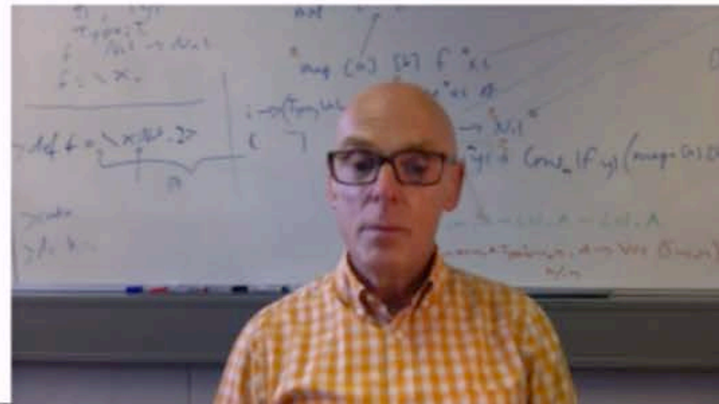
What could go wrong?

To spawn a process on another node, the code for the module needs to be available **and compiled** on the remote node … code isn't mobile!

To spawn a process on another node:

```
Pid = spawn('ant@baz',M,F,A)
```

To name a process on another node, need for the name to be registered at that node.



University of Kent

The small print

Nodes are by default fully connected, but …

… nodes can be hidden, and connected to others "manually" one by one.

Local runtime systems can be stopped and started with `net_kernel` utilities.

Can change cookies dynamically by calling `erlang:set_cookie`

To find out info on yourself and other visible nodes use `node/0` and `nodes/0`.

Further info in the documentation.


University of
Kent