Johan Lindfors

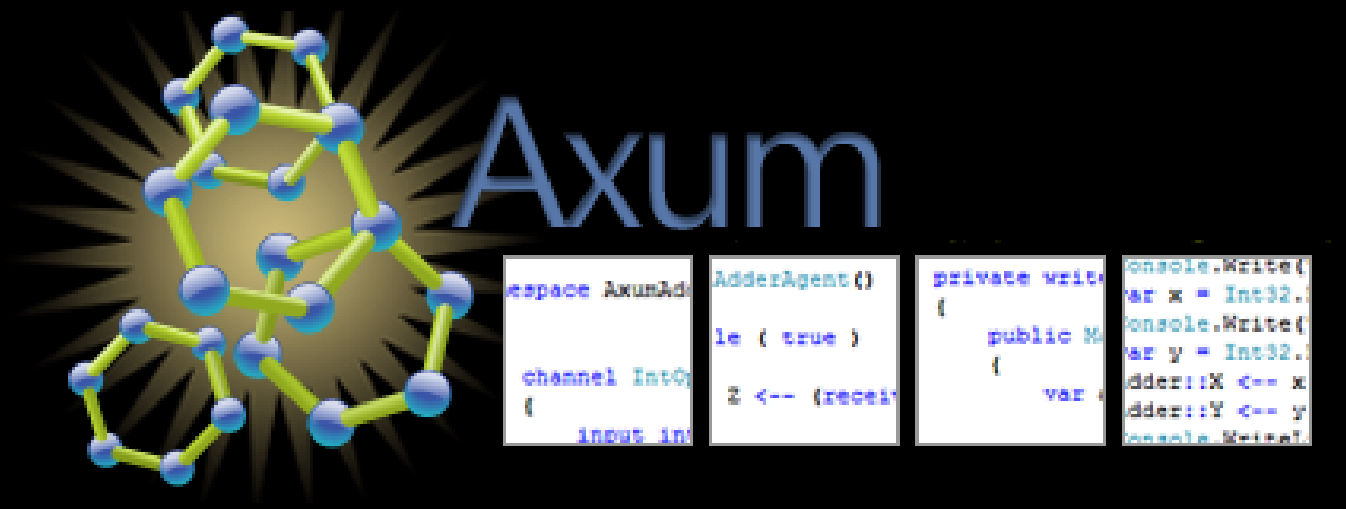
1973

1982

HÖGSKOLAN
DALARNA

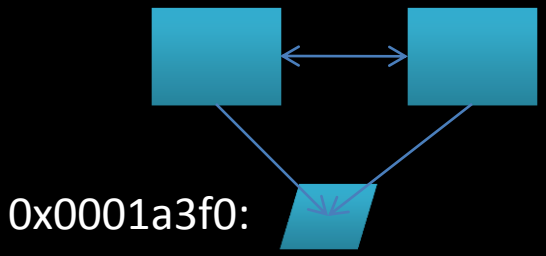
1995

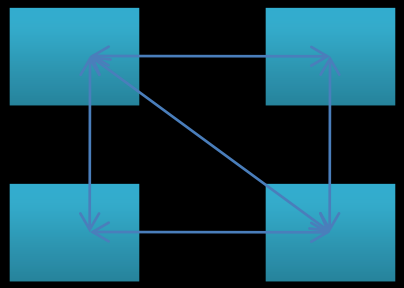
1998

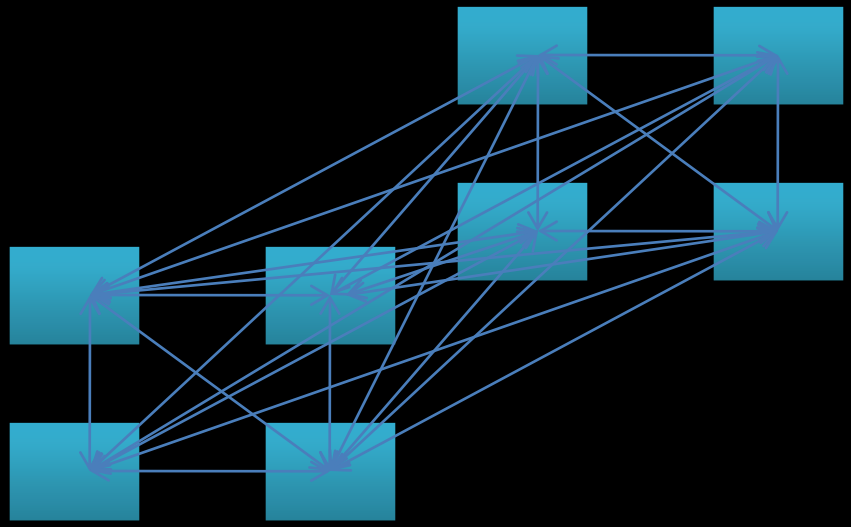

Axum
espace AxumAd
channel IntO
{
input in
AdderAgent()
le (true)
Z <-- (recei
private writ
{
public M
{
var
onsole.Write(
ar x = Int32.
onsole.Write(
ar y = Int32.
dder::X <-- x
dder::Y <-- y

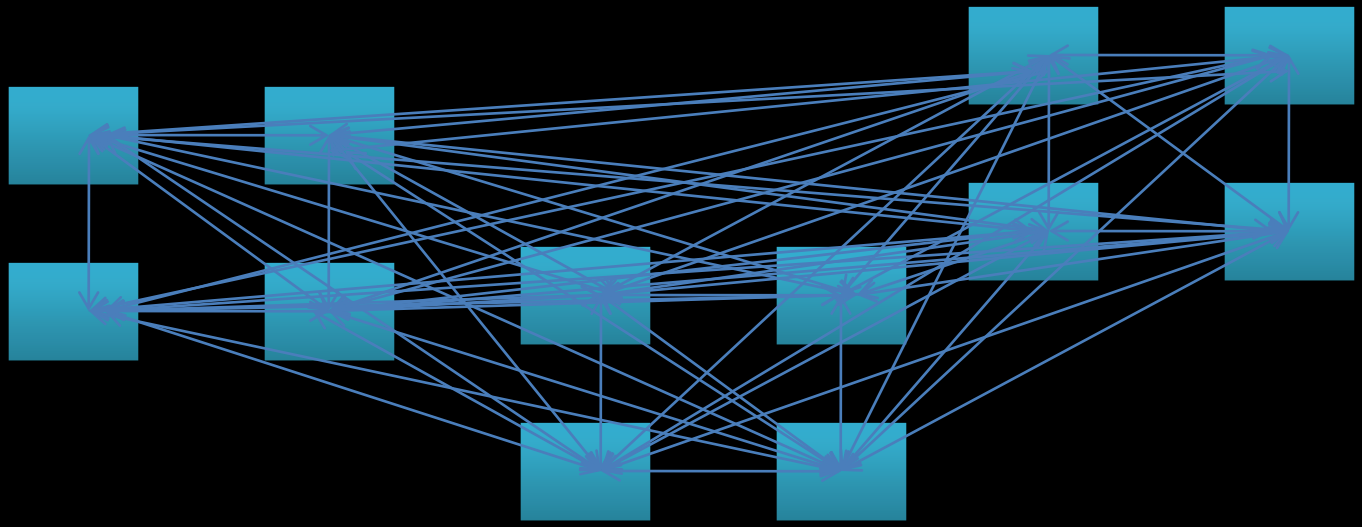
version 0.2

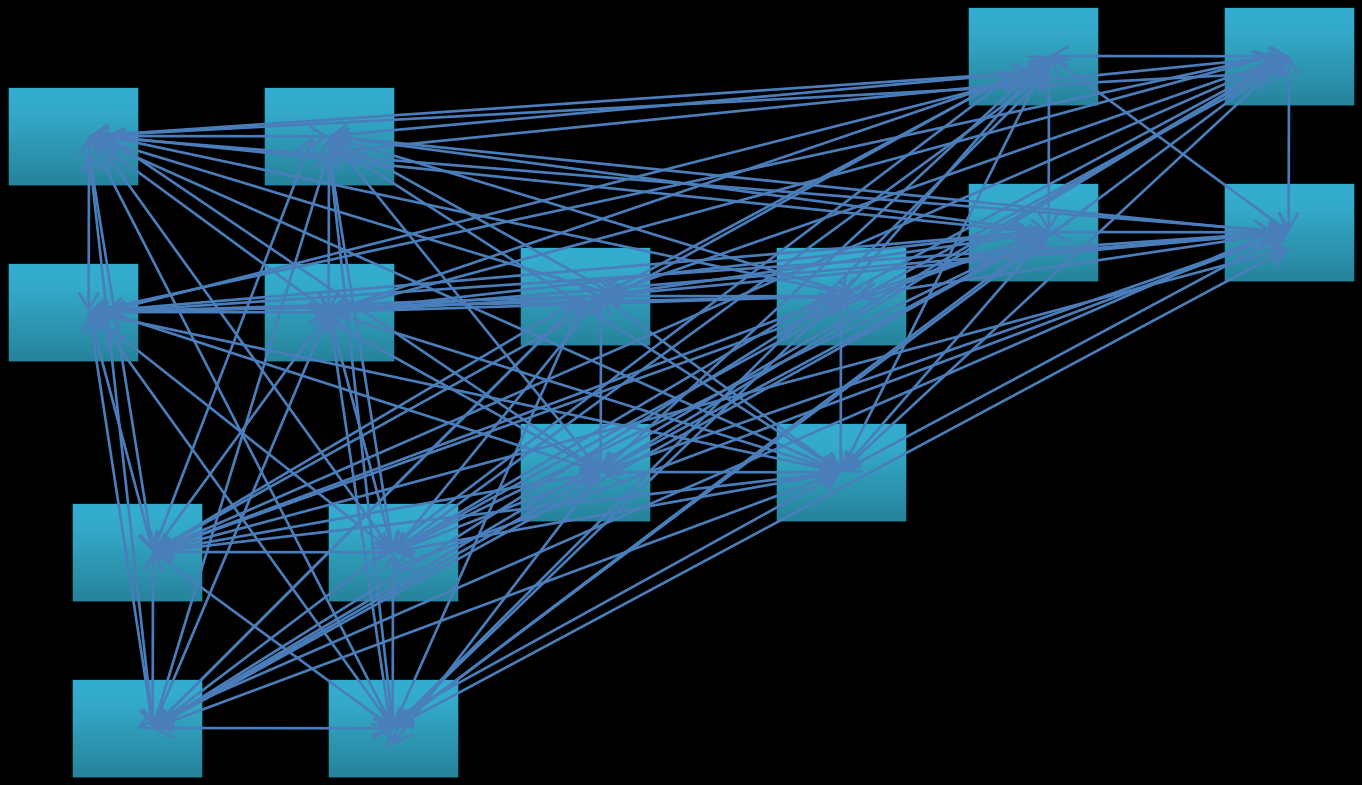
Parallelism is here

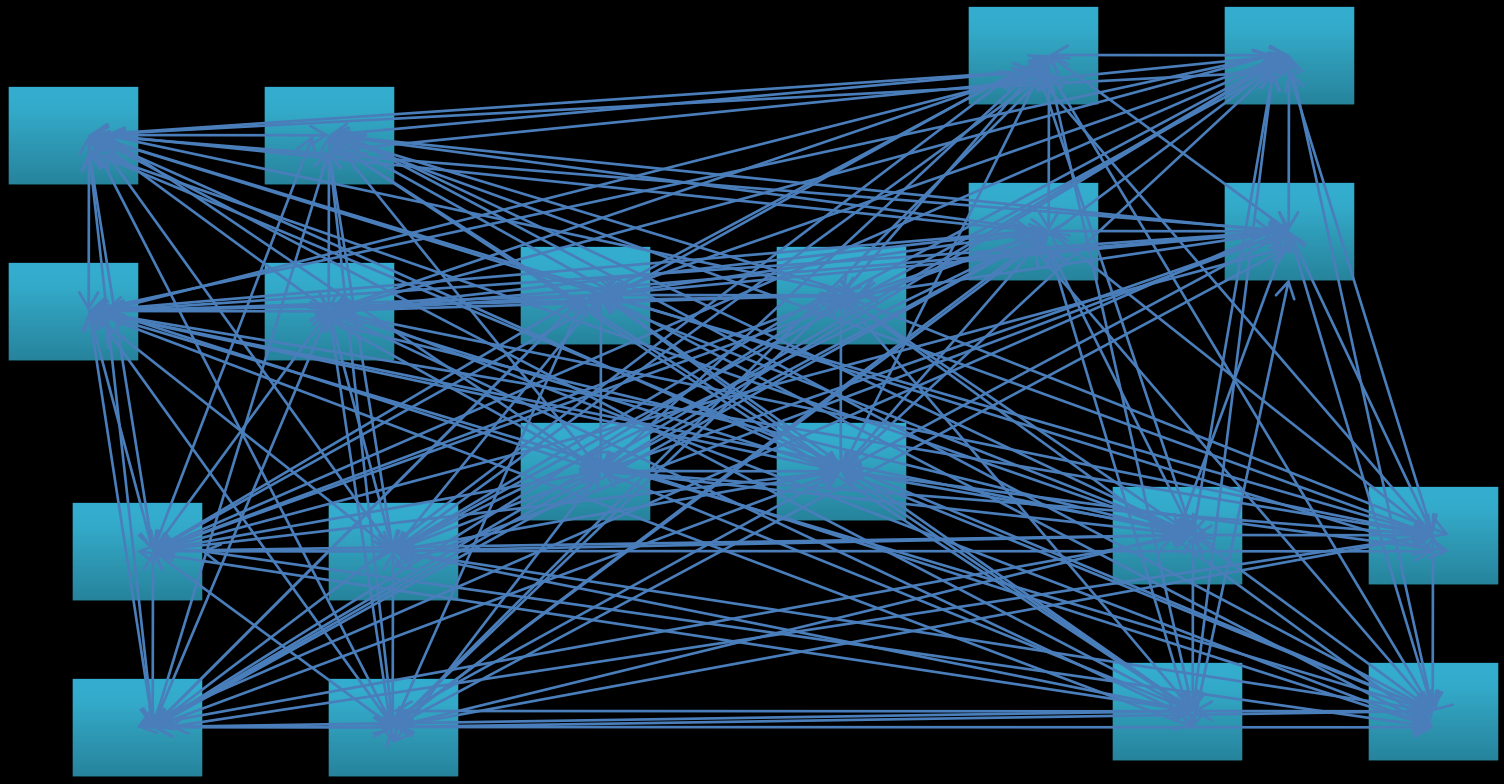

0x0001a3f0:











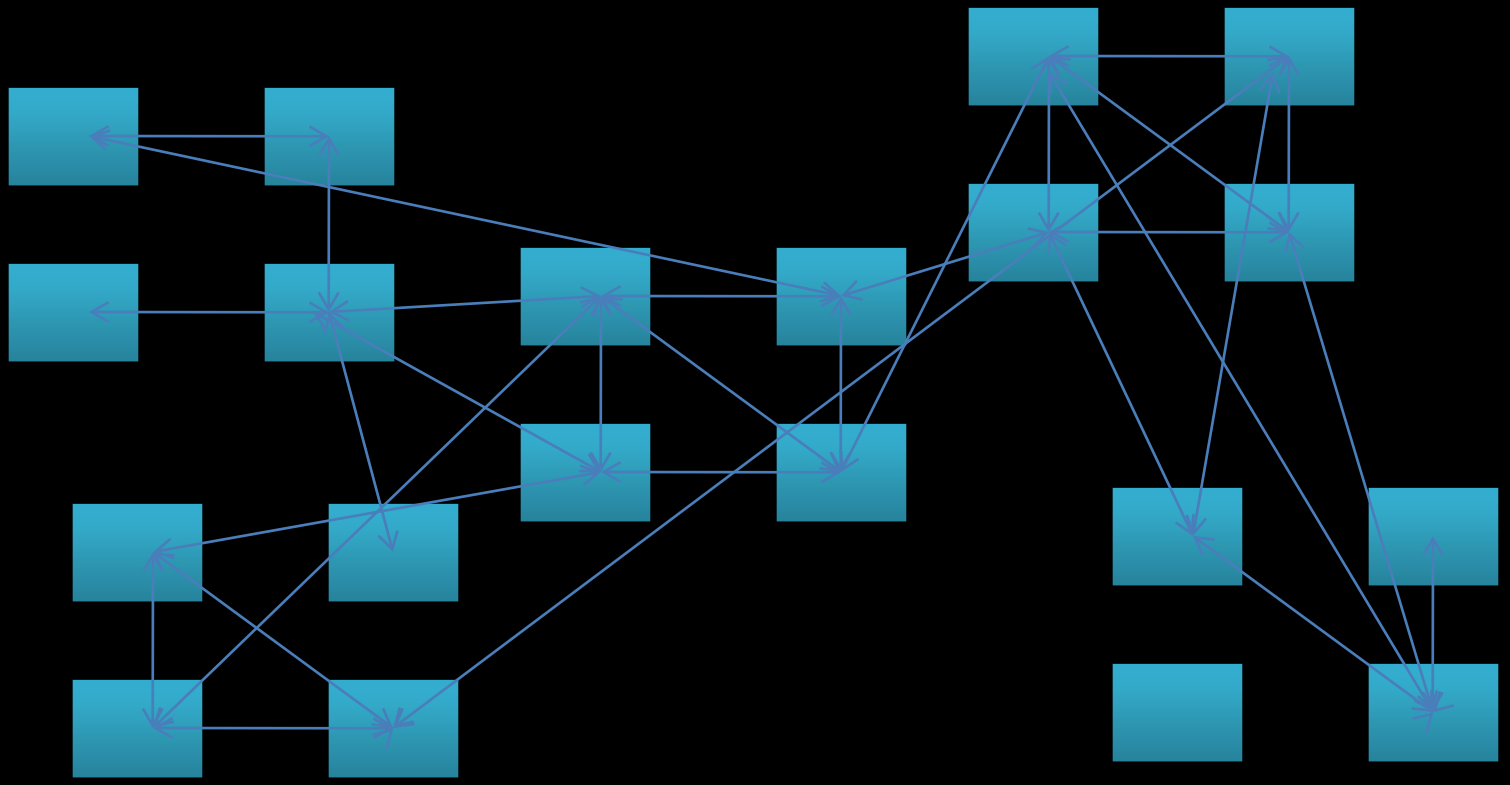

http://www.

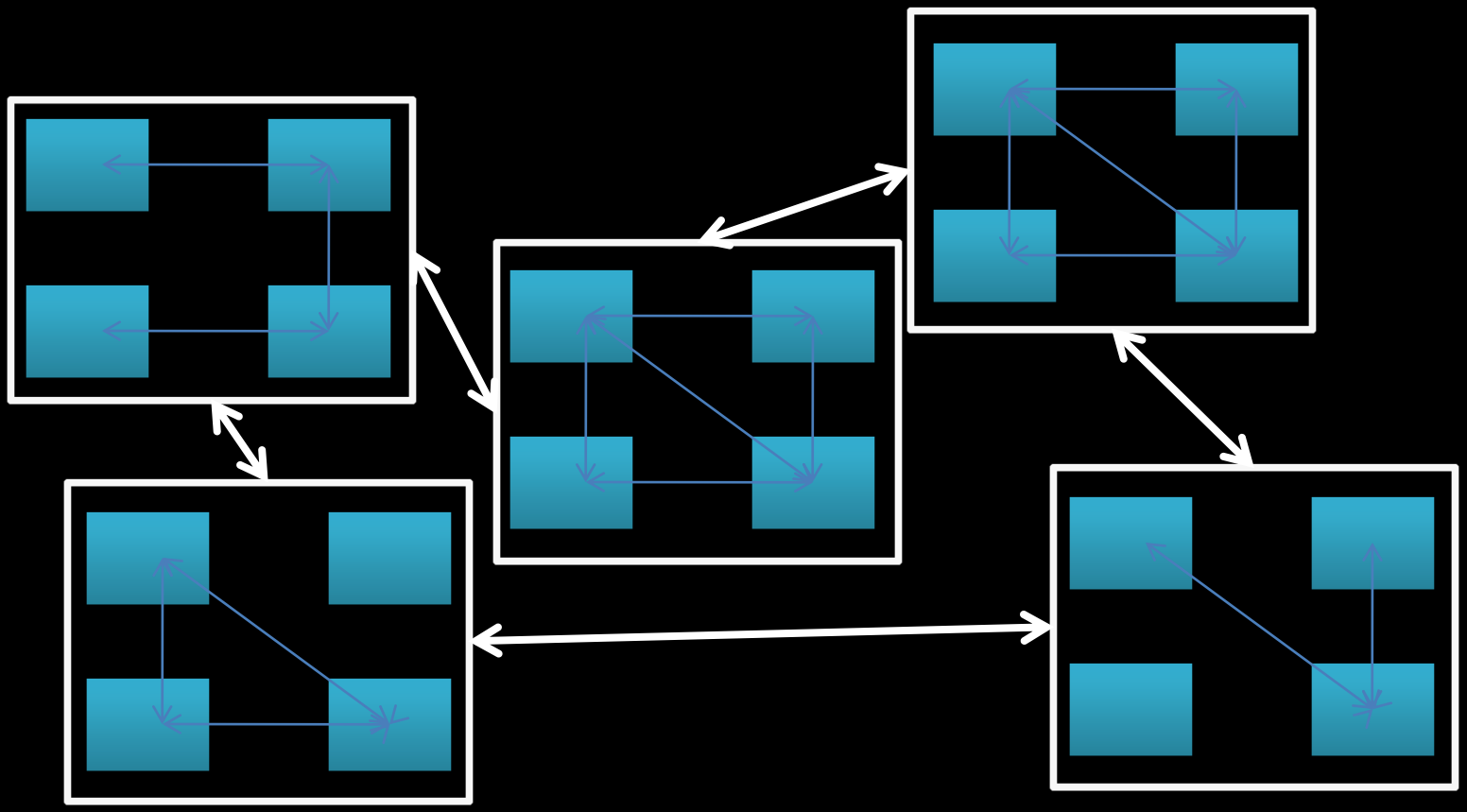
Loosely coupled

Low trust between “components”

Partitioned state

Message-passing (HTTP Get/Put)





Special-purpose language in incubation

Partition data into domains

Agents use shared state within domains

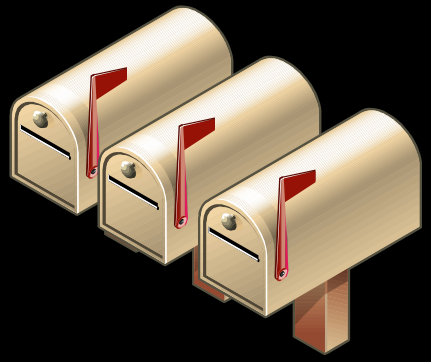
Agents use message-passing between domains

Support for asynchrony and data-flow

Agent

Domain

Channel



Schema

msdn.com/devlabs

msdn.se/johan

Microsoft®

```
channel PingPong
{
    input  bool   Ping;
    output Signal Pong;
}
```

```
agent PingAgent : channel PingPong
{
    public PingAgent ()
    {
        while (receive(Ping))
        {
            Pong <-- Signal.Value;
        }
        Pong <-- Signal.Value;
    }
}
```

```
domain Table
{
    Dictionary<string, string> dict =
          new Dictionary<string,string>();

    public Table()
    {
        Host<TableAgent>(“TableAgentAddress");
    }

    public agent TableAgent : channel TableAccess ...
}
```

```
channel TableAccess
{
    input KeyValuePair <String,String> Put;
    input String Get : String;
    input Signal Done;

    Start: {
          Put $ (!String.IsNullOrEmpty(value.Key)) -> Start;
          Get $ (!String.IsNullOrEmpty(value)) -> Start;
          Done -> End; }
}
```

```
chan::Request ==> TranformString ==> chan::Reply;

buffer -<< { TransformString ==> sink, PrintString };
```

```
schema TableEntry
{
    required String Key;
    required String Value;

    rules { require !String.IsNullOrEmpty(Key);
            require !String.IsNullOrEmpty(Value); }
}
```