THE UNIVERSITY *of* EDINBURGH
informatics

A Declarative Approach to Automated System Configuration

John A. Hewson, Paul Anderson
University of Edinburgh

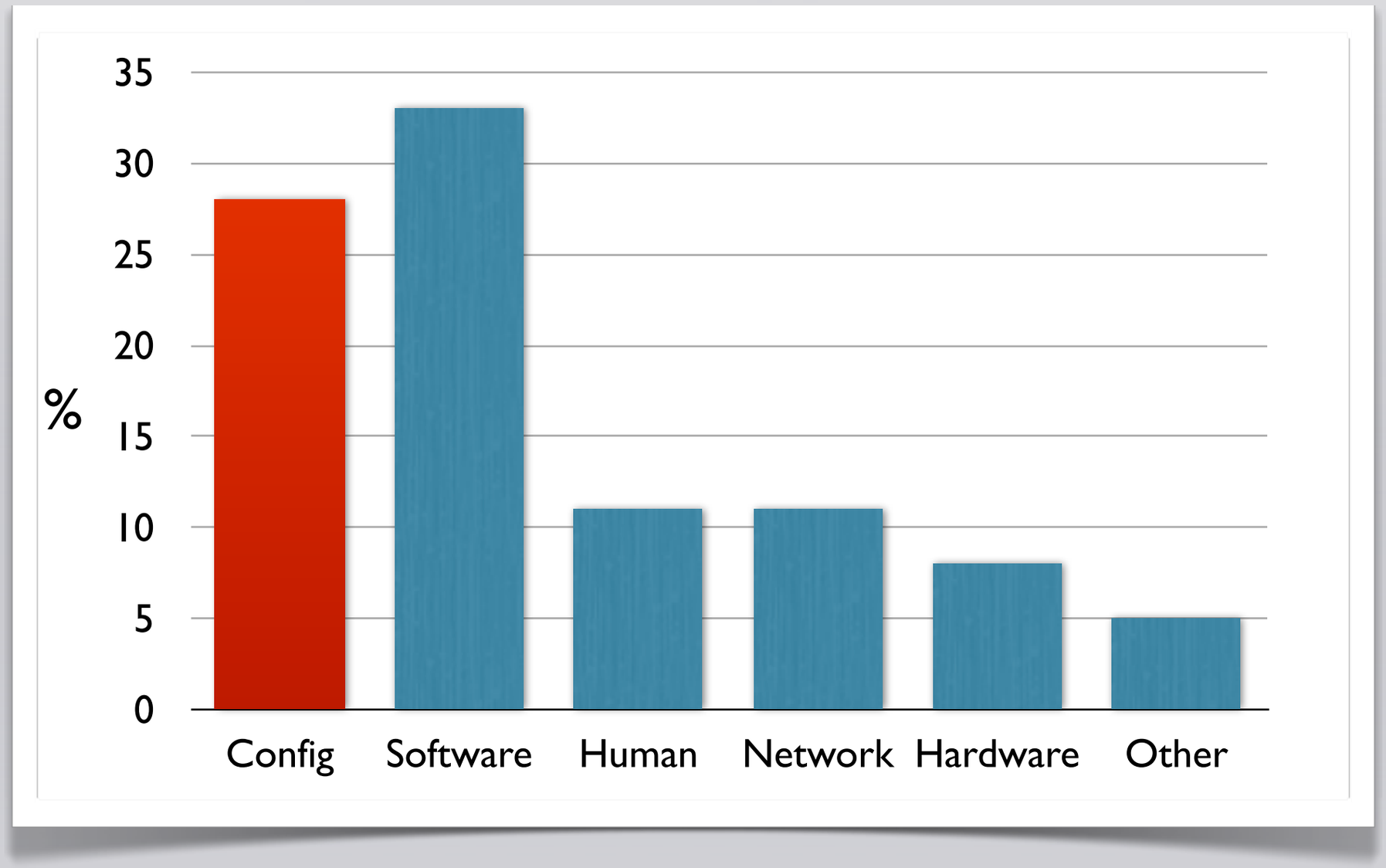
Andrew D. Gordon
Microsoft Research & University of Edinburgh

This work was funded by Microsoft Research through their European PhD Scholarship Program.

Microsoft® Research

Configuration is Hard

Service disruption events by most likely cause at one of Google's main services, over 6 weeks (2009)



The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines, Hoelzle & Barroso, 2009.

What is System Configuration?

Configuring: physical machines, firewalls, networks, datacenters, applications.

Security: proving some invariants hold over both manually and automatically generated configurations.

Cloud: systems are large and force automation

Enterprise systems: are often very complex.

What is Declarative Configuration?

```
package {'apache':
   ensure => installed
}
```

instead of...

```
sudo apt-get -y install apache
```

Declarative Configuration Tools

- **LCFG**, Anderson, 1993, University of Edinburgh
- **CFEngine**, Burgess, 1993, University College Oslo
- **Bcfg2**, Desai, 2004, Argonne National Laboratory
- **Puppet**, Reductive Labs, 2005

But we'd like to...

The ability to **verify** that a configuration conforms to a model

The ability to **infer** valid configurations from a model
How do we...

a) automatically find solutions?

b) write down the models in the first place?

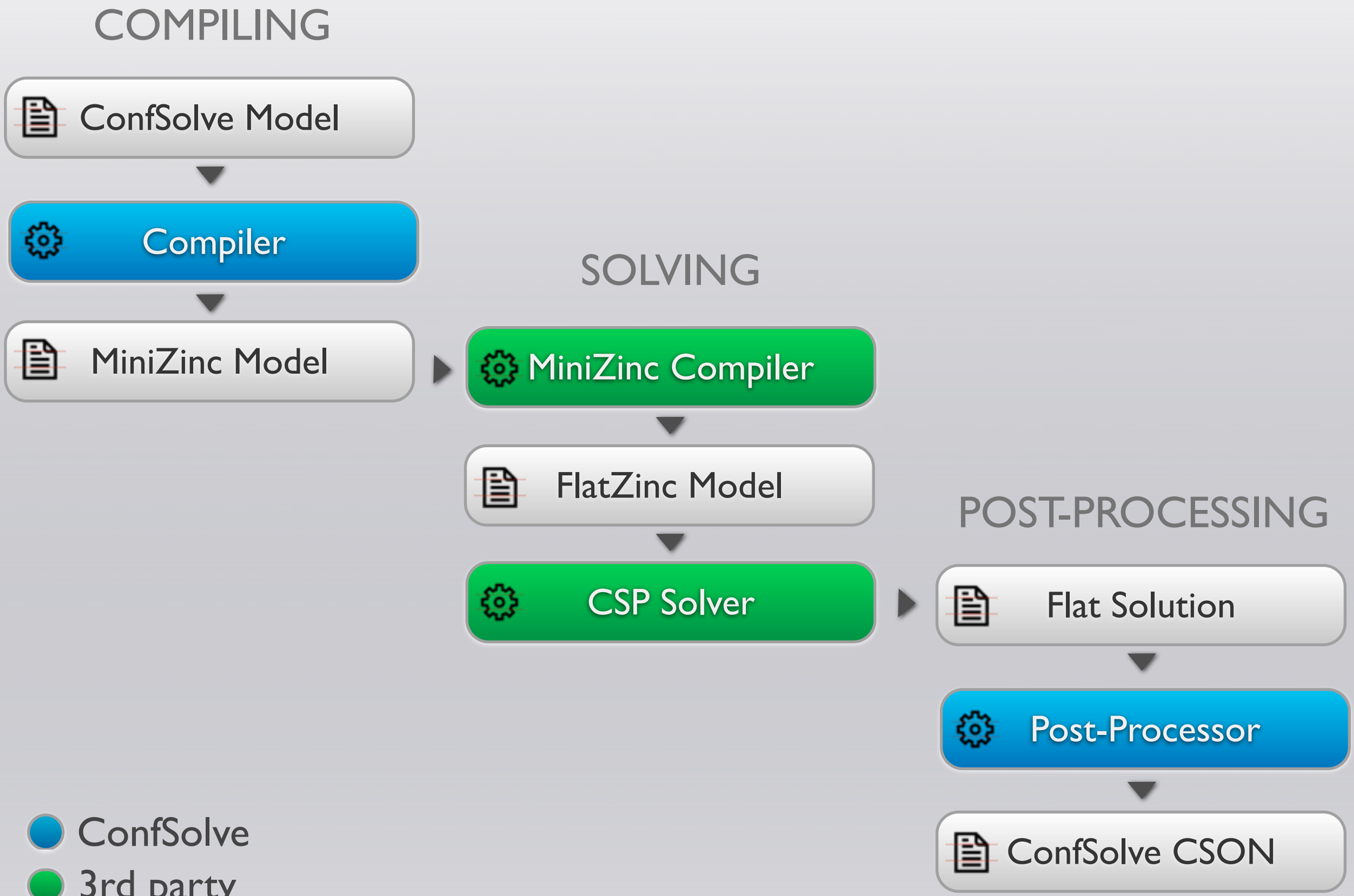
THE UNIVERSITY of EDINBURGH
informatics

ConfSolve

a) automatically find solutions?

ConfSolve: Architecture

THE UNIVERSITY of EDINBURGH informatics



THE UNIVERSITY of EDINBURGH
informatics

ConfSolve

b) write down the models in the first place?

The ConfSolve Language

- designed to be high-level and more familiar to system administrators:
- object oriented (like Puppet, CIM)
- inheritance
- primitives: integer, booleans, sets, enums
- objects, object references, sets of object references
- quantification and summation over decision variables

Example: VMs (1)

```
enum Network { Public, Private }

class Machine {
  var cpu as int
  var memory as int
  var disk as int
  var network as Network

  cpu = 8           // 8 cores
  memory = 16384    // 16 GB
  disk = 2048       // 2 TB
  network = Network.Public
}

class VM {
  var host as ref Machine
  var disk as int
  var cpu as int
  var memory as int
  var network as Network
}
```

Example: VMs (2)

```
class SmallVM extends VM {
  cpu = 1
  memory = 1024  // 1GB
  disk <= 20     // 20GB
}

class LargeVM extends VM {
  cpu = 4
  memory = 4096  // 4GB
  disk <= 500    // 500GB
}
```

Example: VMs (3)

```
var machines as Machine[2]

var sql_server as LargeVM
sql_server.disk = 412

var web_server as SmallVM
web_server.disk = 15
web_server.network = Network.Public
```

Example: VMs (4)

1

```
var vms as ref VM[2]

where foreach (m in machines) {
  sum (vm in vms where vm.host == m) {
    vm.cpu
  } <= m.cpu
```

2

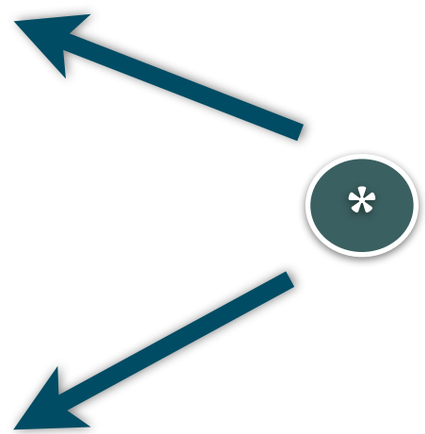
3

```
class Role {
  var host as ref Machine
```

```
  sum (vm in vms where vm.host == m) {
    vm.memory
  } <= m.memory

  sum (vm in vms where vm.host == m) {
    vm.disk
  } <= m.disk
}
```

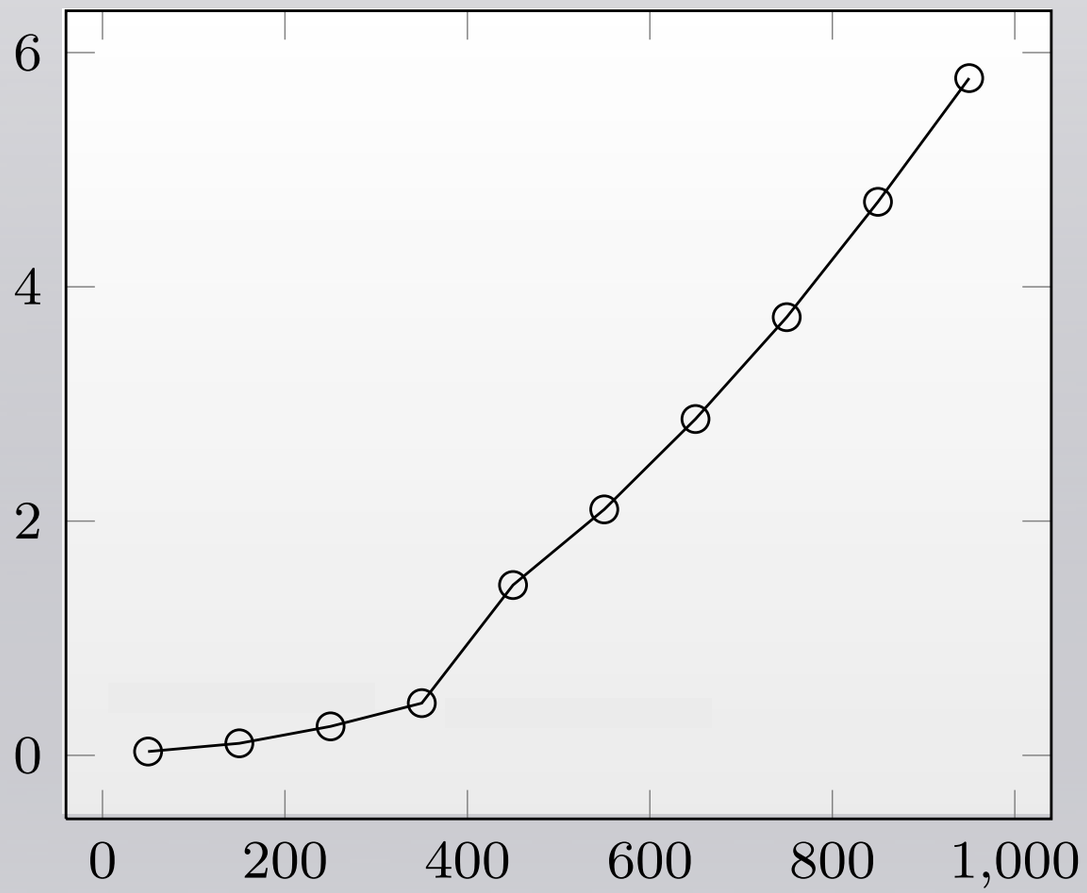
Solution (CSON)

```
vms: VM {sql_server, web_server}

machines[1]: Machine {
    cpu: 16;
    memory: 16384;
    disk: 2048;
    network: Public;
}

machines[2]: Machine {
    cpu: 16;
    memory: 16384;
    disk: 2048;
    network: Public;
}

sql_server: LargeVM {
    disk: 412;
    cpu: 4;
    memory: 3584;
    network: Public;
    host: machines[1];
}

web_server: SmallVM {
    disk: 15;
    cpu: 1;
    memory: 768;
    network: Public;
    host: machines[1];
}
```

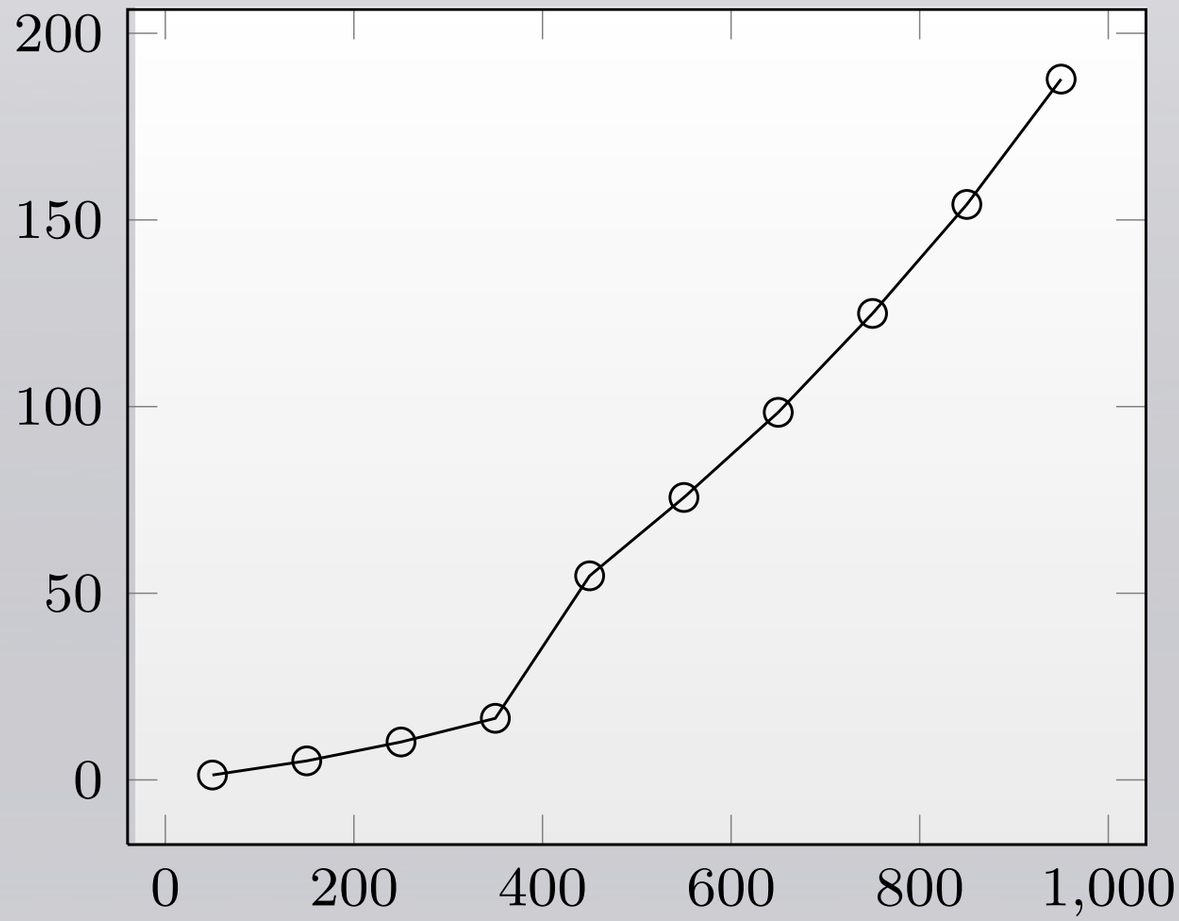


Performance

Memory (GB)



Time (sec)

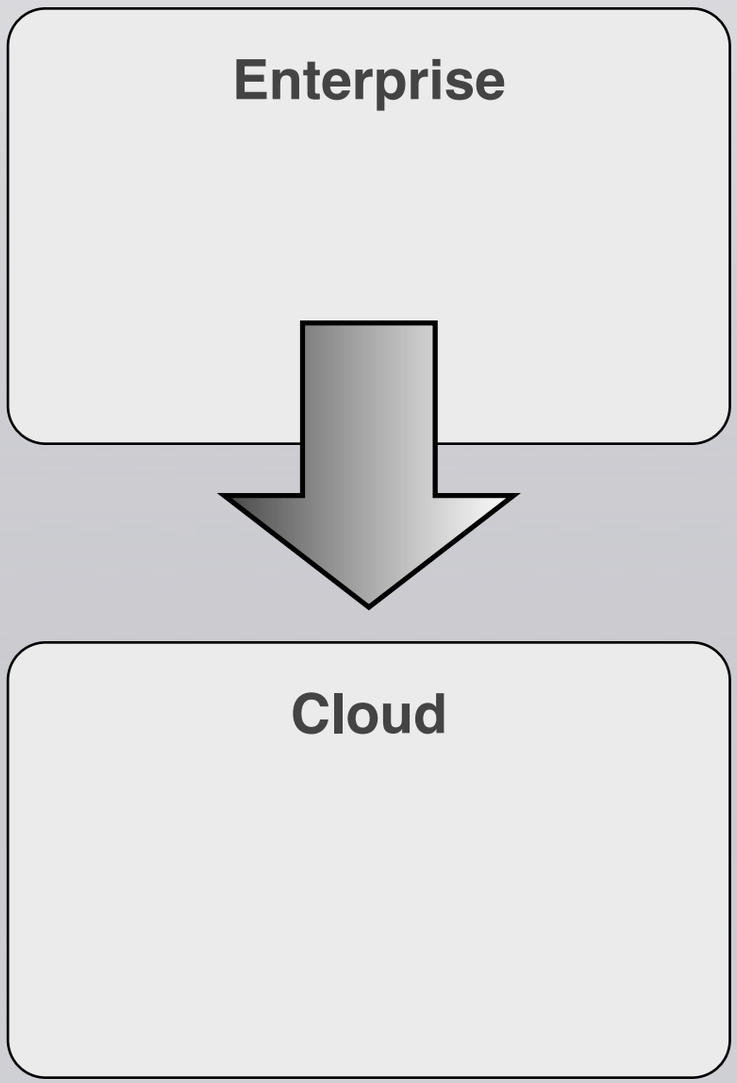


Optimization

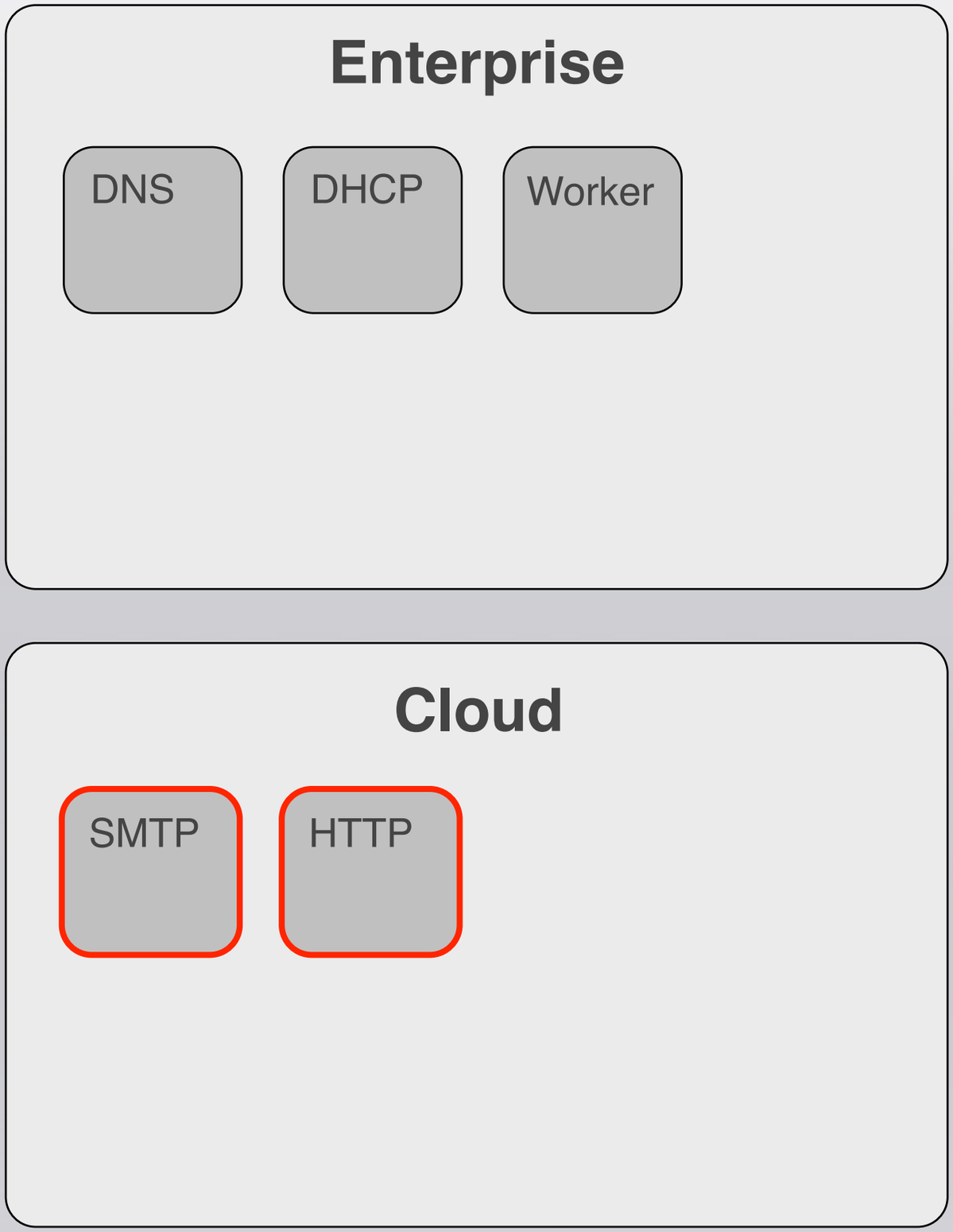
Optimization

- Often want to optimize some aspect of the configuration
- or express *soft* preferences rather than *hard* constraints.
- The underlying solver supports maximization of an objective function.
- For ConfSolve this is not just useful, but essential...

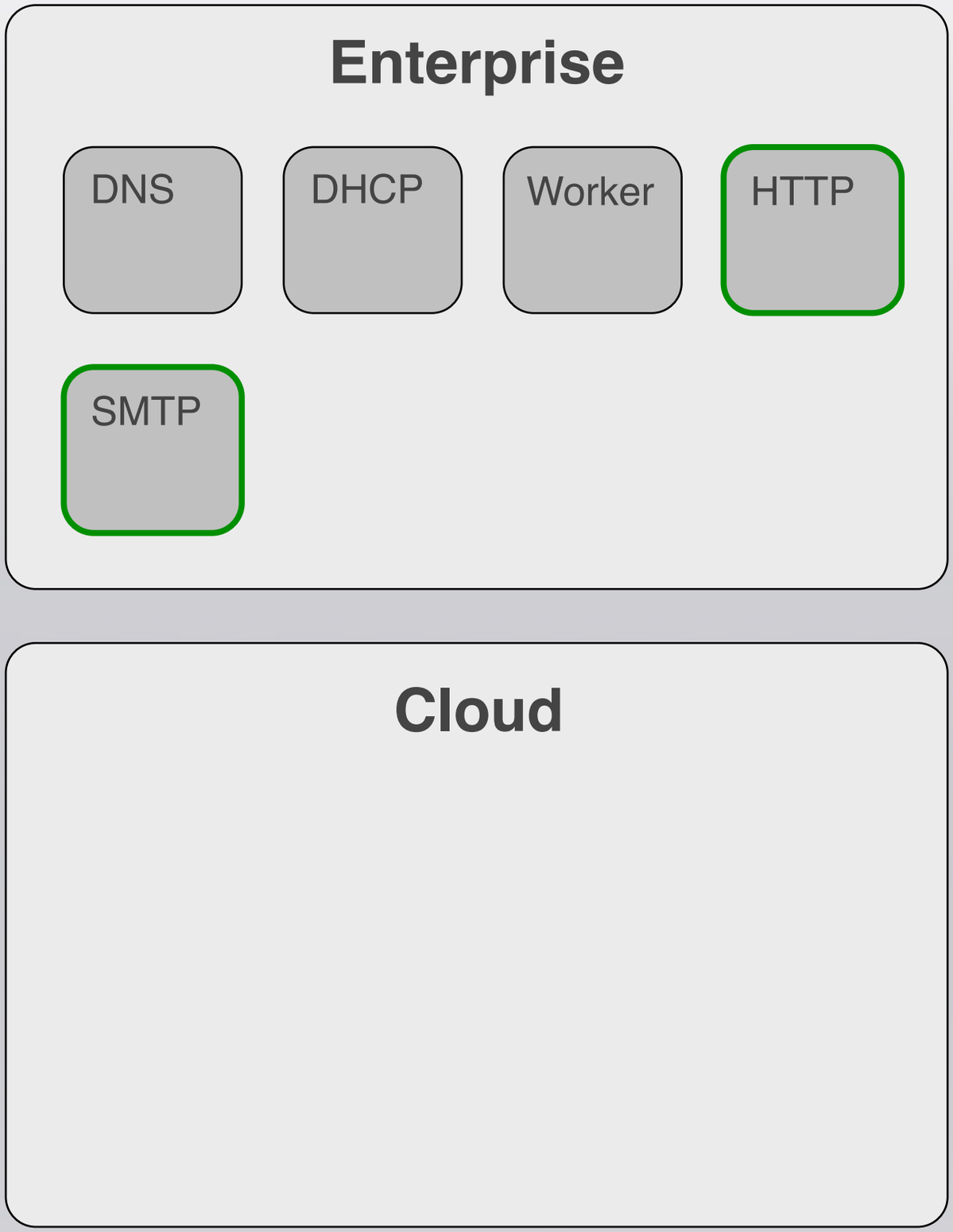
Example: *Cloudbursting*



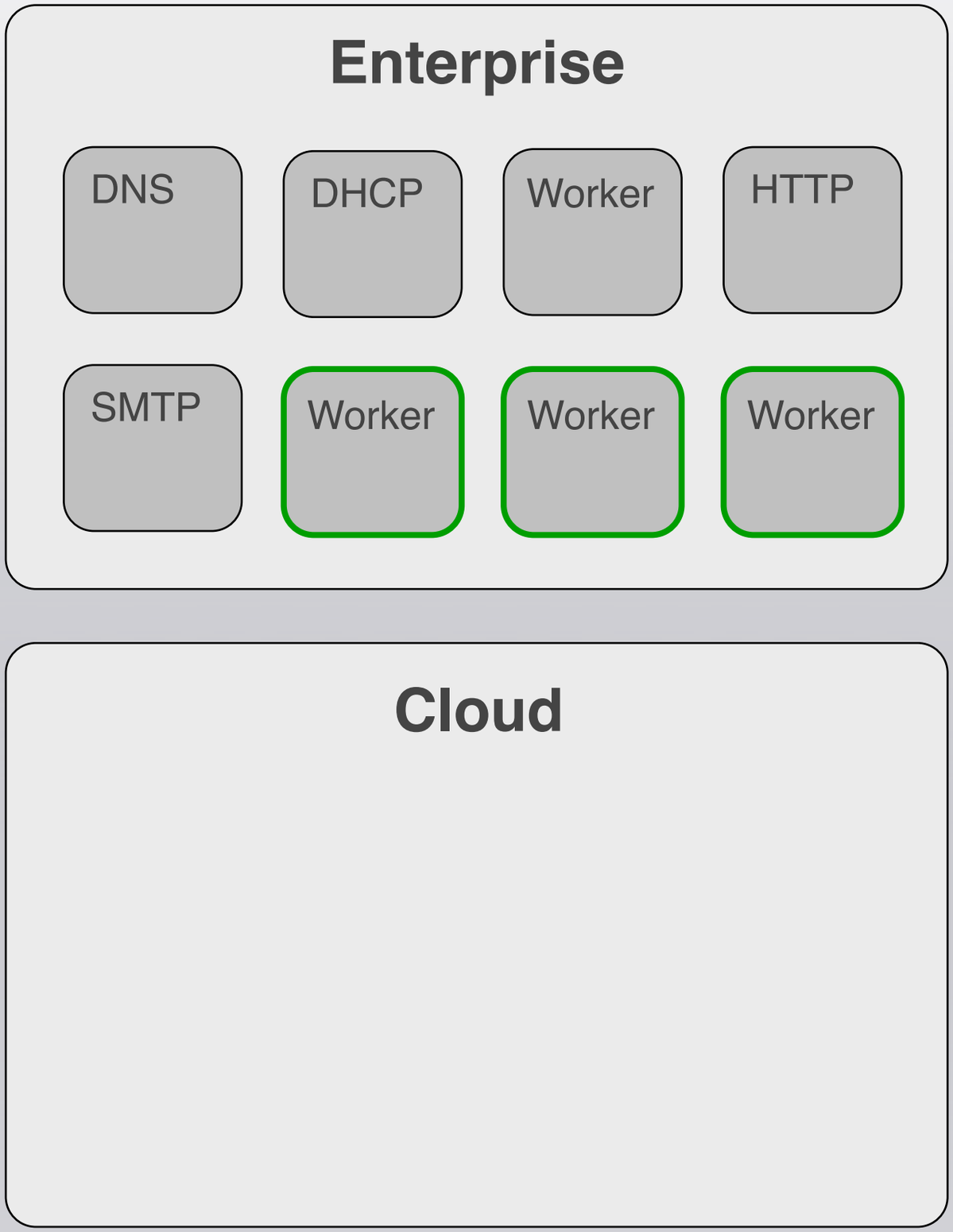
Without Optimization



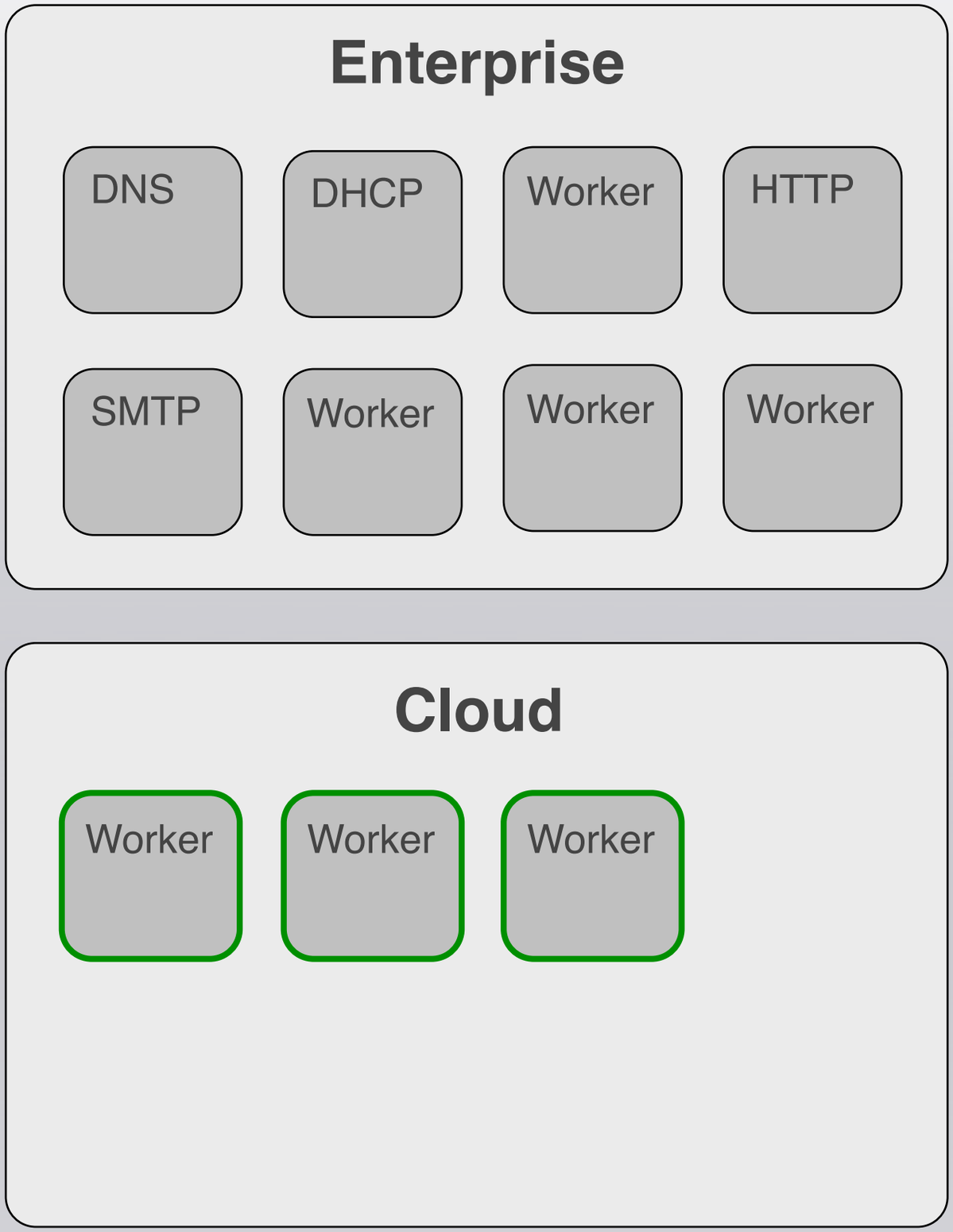
With Optimization



With Optimization



With Optimization



Example: Cloudbursting

THE UNIVERSITY of EDINBURGH informatics

3

```
class Machine;

class Service {
   var host as ref Machine;
}

class Datacenter {
   var machines as Machine[8];
}
```

2

```
var cloud as Datacenter;
var enterprise as Datacenter;
```

3

```
var dhcp as Service[1];
var dns as Service[1];
var workers as Service[3];
```

4

```
maximize count (s in services
                where s.host in enterprise.machines);
```

Future Work

- Reconfiguration
  - What happens the 2nd time we configure a system, or the 3rd, 4th, 5th?
  - How do we react to changes but minimize impact?

- ConfSolve provides a platform which could be used to augment existing configuration languages

THE UNIVERSITY of EDINBURGH
informatics

Thank You

http://homepages.ed.ac.uk/s0968244/

john.hewson@ed.ac.uk

@jahewson