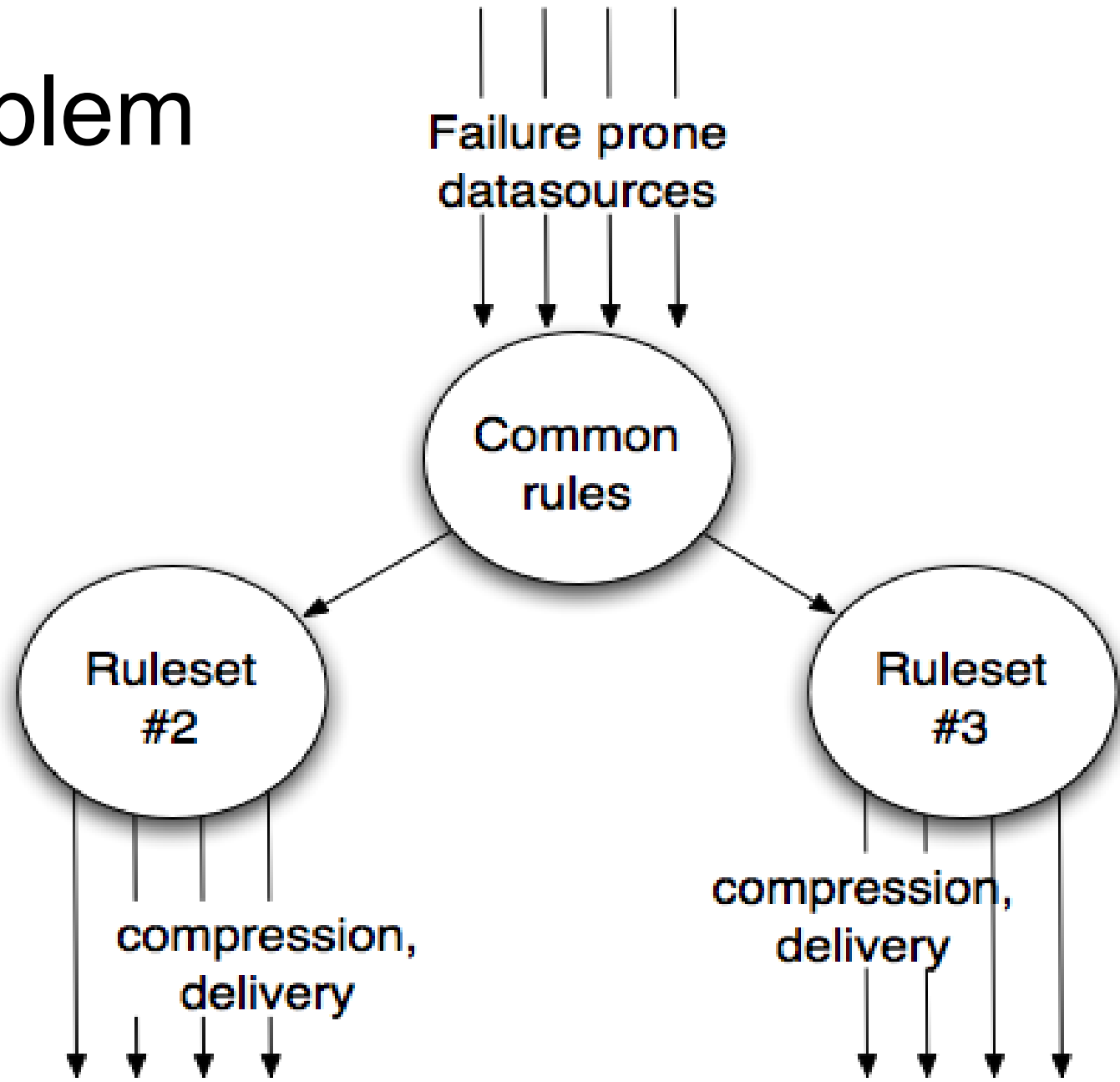


The problem



The solution

... some clean way that I can
set up a hierarchy of my
business rules

In my case, this stuff should Just Work

- Log::Log4perl (woot!)
- Working directory, archive directory, filename defaults
- Scheduling
- Fork maximum of X children at a time (queue the rest)
- Child timeouts
- Child result aggregation
- Encryption
- Delivery
- Notification
- Retries

*Let me focus on the
exceptions only.*

Cabal of Awesome

POE

- Stable. Too stable!!
- Event handling
- Child forking
(POE::Wheel::Run)
- Signal handling

Moose

- Inheritance / Roles to any level of business rule complexity
- DRY
- Clean 'n pretty

POE::Wheel::Run + Moose = MooseX::Workers

```
package Manager;  
use Moose;  
with qw(MooseX::Workers);  
  
sub run {  
    $_[0]->spawn( sub { print "Hello World\n" } );  
    POE::Kernel->run();  
}  
sub worker_stdout { warn $_[1]; }  
  
Manager->new->run();
```

Callbacks

```
sub run { }
```

```
sub worker_manager_start { warn 'started worker manager' }
```

```
sub worker_manager_stop { warn 'stopped worker manager' }
```

```
sub max_workers_reached { warn 'max worker count reached' }
```

```
sub worker_started { }
```

```
sub worker_stdout { }
```

```
sub worker_stderr { }
```

```
sub worker_error { }
```

```
sub worker_done { }
```

```
sub worker_timeout { }
```

```
sub sig_child { }
```

enqueue()

```
sub run {  
    for my $num (1..3) {  
        $_[0]->enqueue(  
            sub { print "HELLO $num\n" }  
        );  
    }  
    POE::Kernel->run();  
}
```

```
my $Manager = Manager->new();  
$Manager->max_workers(2);  
$Manager->run();
```

MooseX::Workers::Job

```
sub run {  
    foreach (qw( foo bar baz )) {  
        my $job = MooseX::Workers::Job->new(  
            name      => $_,  
            command => sub { print "Hello World\n" },  
            timeout => 30,  
        );  
        $_[0]->spawn( $job );  
    }  
    POE::Kernel->run();  
}
```


MooseX::Workers::Job

```
sub worker_stdout {  
    my ( $self, $output, $job ) = @_  
    printf(  
        "%s(%s,%s) said '%s'\n",  
        $job->name, $job->ID, $job->PID, $output  
    );  
}  
Manager->new()->run();  
# bar(2,32423) said 'Hello World'  
# foo(1,32422) said 'Hello World'  
# baz(3,32424) said 'Hello World'
```

TO DO

- Suggested yesterday:
Add automatic retries, `max_retries()` setting
- Push more of my `$work` namespace code down into `MooseX::Workers`?
- <http://github.com/perigrin/moosex-workers>
or [/jhannah/](#)
- ...requests? jay@jays.net
[jhannah](#) on [irc.perl.org](#) `#moose` `#perl-help` etc.