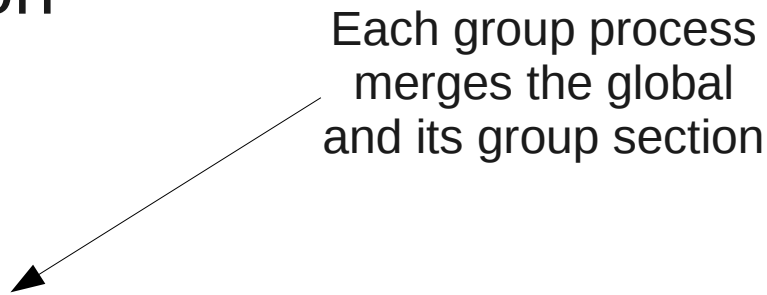
glideinWMS Frontend Configuration

by Igor Sfiligoi (UCSD)

Configuration file structure

- The Frontend configuration file is an XML file
  - With **`<frontend/>`** as the top element
  - Nothing special about it
- Composed of
  - An instance description section
  - A global section
  - A group list section
    - Each group has its own section mirroring the global section structure



http://tinyurl.com/glideinWMS/doc.prd/frontend/configuration.html

Description section

Description section

- It can be further divided into:
  - Base directory and network setting
  - Versioning
  - Own security settings
  - Interface to Central Manager
  - Optimization knobs

Directories and networking

- You should get this out of the installer

Example section

```
<work base_dir="/home/frontend/frontstage"
        base_log_dir="/home/frontend/frontlogs"/>

<stage base_dir="/var/www/html/vofrontend/stage"
        use_symlink="True"
        web_base_url="http://frontend.node/vofrontend/stage"/>

<monitor base_dir="/var/www/html/vofrontend/monitor"
          flot_dir="/home/frontend/javascriptrrd-0.6.1/flot"
          javascriptRRD_dir="/home/frontend/javascriptrrd-0.6.1"
          jquery_dir="/home/frontend/javascriptrrd-0.6.1/flot"/>
```

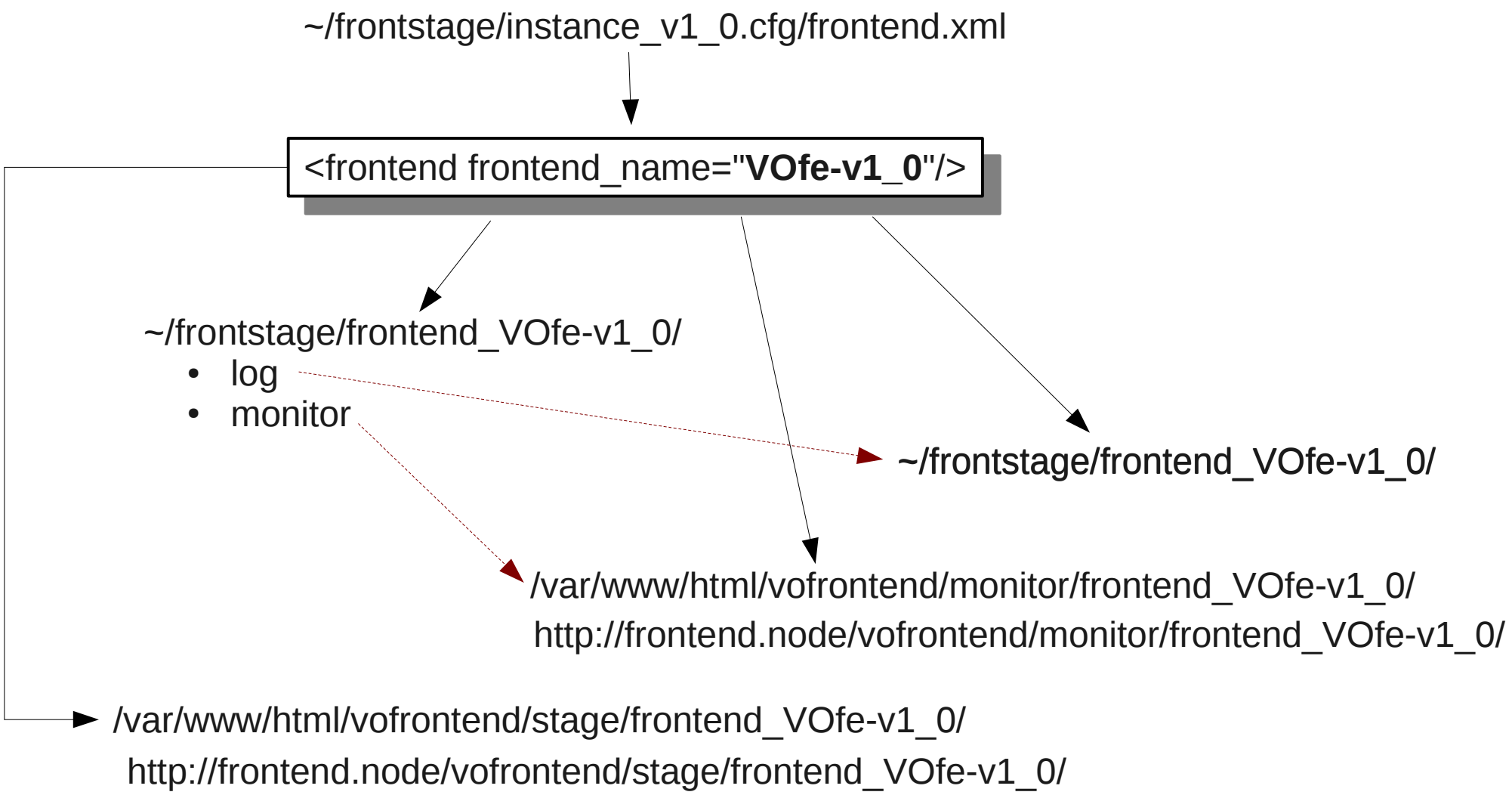
Versioning

- The Frontend has its own name
  - Which may or may not be related to the Security Name
- This name is used for
  - Request ClassAd Name (toward the Factory)
  - Creating the actual directory structure
- By adding a version string, easy to upgrade without losing history

Not in the RPM version

Or you can run several instances in parallel on the same node

Example versioning



Own security and CM

- These also filled out by the installer

Example sections

```
<security security_name="VOsomething"
          classad_proxy="/home/frontend/.globus/x509_service.proxy"
          proxy_DN="DNxxx"
          sym_key="aes_256_cbc"/>

<collectors>
    <collector DN="DNyyy" node="cm.node" secondary="False"/>
    <collector DN="DNyyy" node="cm.node:9620-9719" secondary="True"/>
    ...
  </collectors>
```

Top collector

Leafs of the Collector tree ...
there may be many such lines

Optimization knobs

- Defaults are usually OK

```
<frontend
   advertise_with_multiple="True"
   advertise_with_tcp="True"
   advertise_delay="2"
   loop_delay="60"
   restart_attempts="3"
   restart_interval="1800">
   <log_retention max_days="30.0"
                  max_mbytes="4000.0"
                  min_days="7.0"/>
</frontend>
```

How to talk to the Factories.
Keep it to True/True!

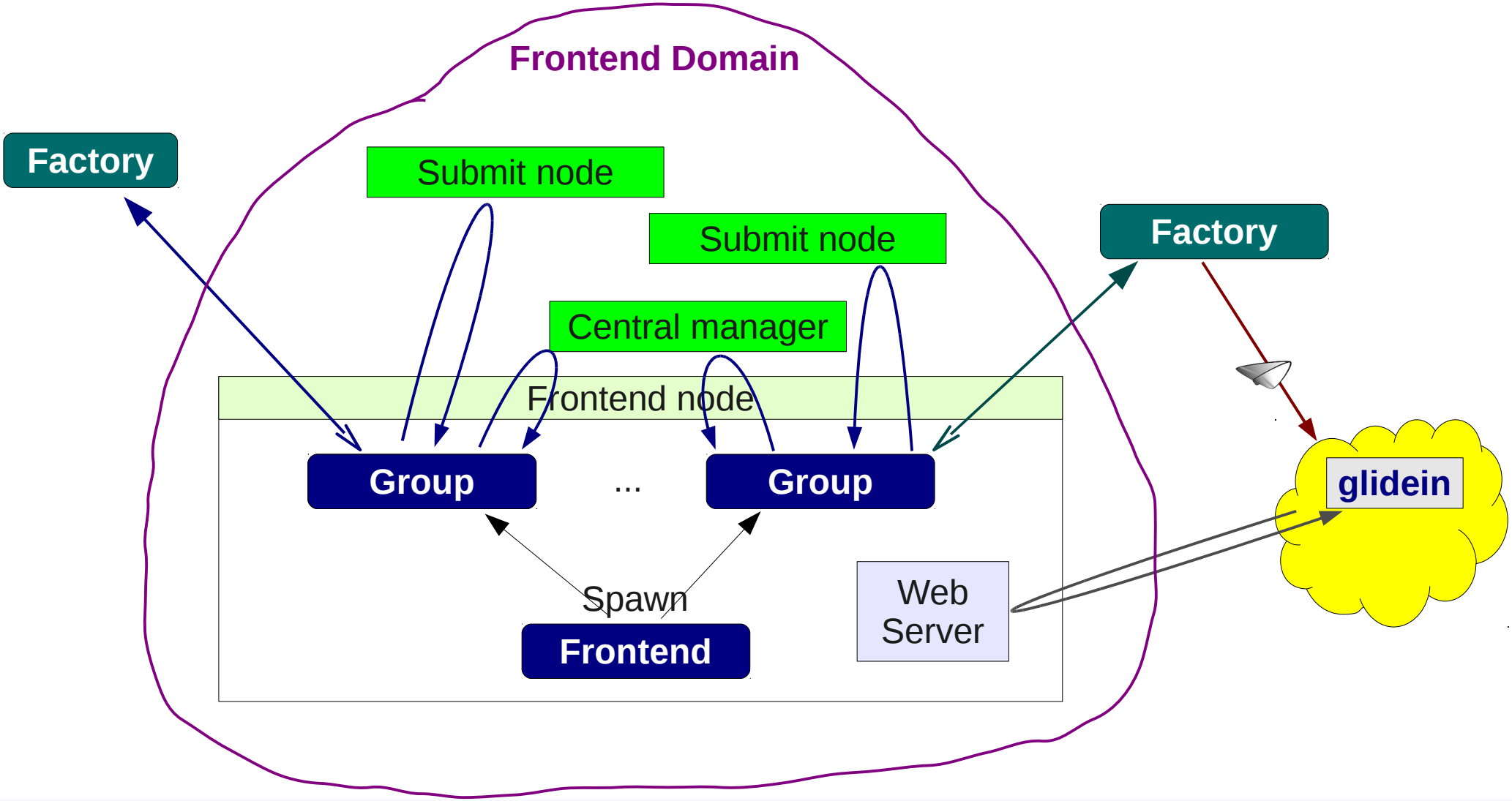
How tight should be the polling loop.

In case something goes wrong.

How long to preserve logs

Global vs Group sections

Refresher – Frontend Groups



Frontend groups

- The Frontend Groups are those who actually do the real work
  - So we will have a section for each group

Not recursive

```
<groups>
  <group name="group1" enabled="True">
    …
  </group>
  ...
</groups>
```

You **must** have at least one.

The installer creates one group called "main".

- The global section is for anything that is in common between groups
  - So we do not need to cut-and-paste

Shared sections

- Both global and group sections contain:
  - A pilot proxy section
  - An attribute section
  - A file section
  - A matchmaking section
- Currently, the group section alone also has:
  - A limits section

The pilot proxy section

- Essentially, you must list the proxies
  - **Both** the global and the group proxies used
- If you have more than one proxy total, you can pick a plugin to select between them
  (plugins not described in this talk)

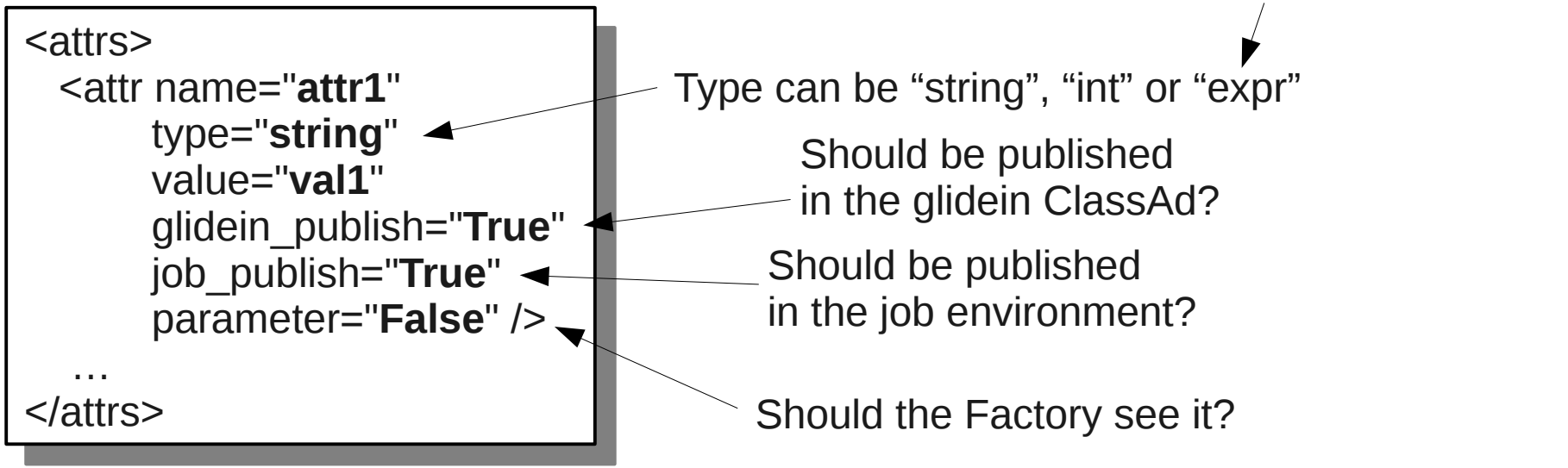
By default, use all proxies

```
<security>
 <proxies proxy_selection_plugin="ProxyAll">
  <proxy absfname="/home/frontend/.globus/x509_vopilot01.proxy"
         security_class="frontend"/>
  ...
 </proxies>
</security>
```

http://tinyurl.com/glideinWMS/doc.prd/frontend/configuration.html#multiple_proxy

The attribute section

- You can provide arbitrary attributes to the glideins
  - Both validation scripts and Condor
- If you define the same attribute both in the global and the group section
  - The group defined one will prevail



Useful attributes

- You may want to set these:
  - GLIDEIN_Job_Max_Time – Expected job lifetime.
  - GLIDEIN_Expose_Grid_Env – Expose grid env to jobs?
  - GLIDEIN_Glexec_Use – Should I use gLExec?
  - GLIDECLIENT_Group_Start – The START expression (more later)
  - GLIDEIN_Max_Idle – Max wallclock time wasted.

http://tinyurl.com/glideinWMS/doc.prd/factory/custom_vars.html

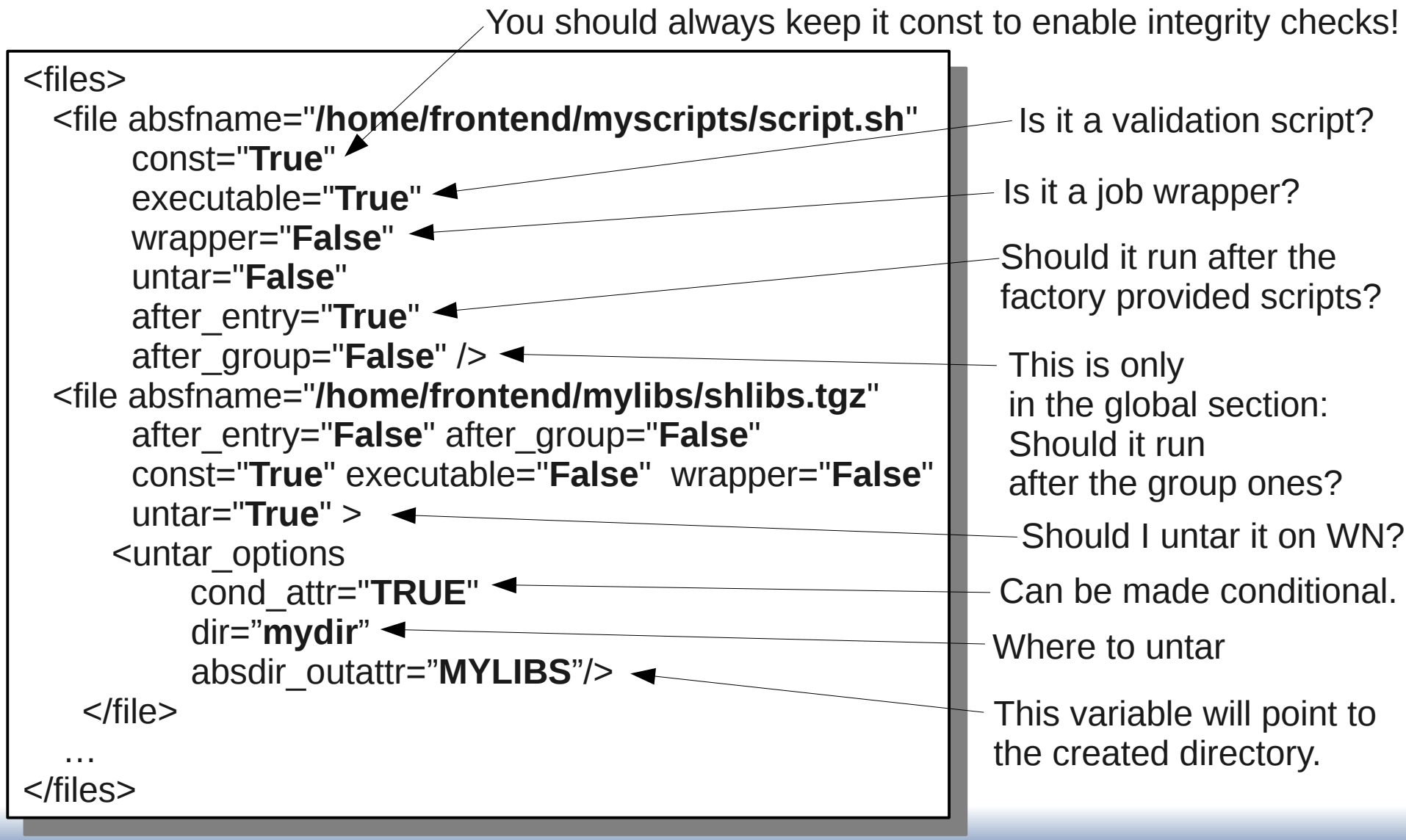
The file section (1)

- Here you can send arbitrary files to the glideins
  - All of the global and the group files will be used
  - In the order in which they are specified
- Example files:
  - Validation scripts
  - Configuration files for the validation scripts
  - VO apps and libs that the users need
  - Wrappers to be run before every user job
    http://research.cs.wisc.edu/condor/manual/v7.6/3_3Configuration.html#param:UserJobWrapper
- Can set attribute to point others to the files
- Can untar a tarball

Global vs group order can be altered (see next slide)

Instead of prestaging

The file section (2)



The limits section

- Specify the limits you never want exceeded
  - Idle limits define how aggressive you want to be
  - Running limits are essentially emergency breaks

```
<config>
    <idle_glideins_per_entry max="100" reserve="10"/>
    <idle_vms_per_entry curb="10" max="50"/>
    <running_glideins_per_entry max="5000" relative_to_queue="1.05"/>
    <running_glideins_total curb="17000" max="20000"/>
</config>
```

The matchmaking section

The matchmaking section

- The matchmaking section itself is composed of
  - The factory selection section
  - The job selection section
  - The actual matchmaking bits
- You will also want to add a related attribute
  - The START expression for Negotiator matchmaking

The factory selection section

- The entry can talk to many Factory collectors
  - And not all Factory entries may be relevant
- Specify Factory Collector params and filters
  - As obtained from the Factory admins

```
<match>
    <factory query_expr='stringListMember("VO",GLIDEIN_Supported_VOs)'>
        <collectors>
            <collector node="factory1.node"
                       DN="DNxxx"
                       factory_identity="gfactory@factory1.node"
                       my_identity="VOyyy@factory1.node" />
            ...
        </collectors>
    <factory>
</match>
```

The job selection section

- Here you specify which jobs to look at
  - And from which schedds
- Useful for
  - Filtering out jobs that will not use glideins
  - Partitioning jobs between different groups

```
<match>
    <job query_expr="(JobUniverse==5)'>
      <schedds>
        <schedd fullname="schedd1.node"
                DN="DNxxx" />
        ...
      </schedds>
    <job>
</match>
```

Matchmaking

- A python expression
  - If evaluates to True, the job matches the factory (entry)
  - glidein["attrs"]["**ATTRxxx**"] – Factory attribute
  - job[“**ATTRyyy**”] - Job attribute

Formatting only:
Splitting across lines
unfortunately not legitimate

```
<match
   match_expr='(job.has_key("DESIRED_Sites") and
                (glidein["attrs"]["GLIDEIN_Site"] in job["DESIRED_Sites"].split(",")))'/>
```

Matchmaking attributes

- You must specify which attributes you will use
  - Both for the factory and the jobs
  - By default you get only the ClassAd names!

```
<match match_expr='glidein["attrs"]["GLIDEIN_Site"] in job["DESIRED_Sites"].split(","))'>
  <factory query_expr="GLIDEIN_Site=!=UNDEFINED”>
    <match_attrs>
      <match_attr name="GLIDEIN_Site" type="string"/>
    </match_attrs>
  </factory>
  <job query_expr="DESIRED_Sites=!=UNDEFINED”>
   <match_attrs>
     <match_attr name="DESIRED_Sites" type="string"/>
   </match_attrs>
  </job>
</match>
```

Valid types are:
string, int, real and bool

Adding a Start Expression

- The match expression will make sure that glideins start
  - Now you must make sure they run only “right” jobs
- i.e. you need to provide a Start expression for the Negotiator to use
- The glidein combines the following attributes with an AND:
  - GLIDECLIENT_Start
  - GLIDECLIENT_Group_Start

} One for the global and one for the group scope. Both default to **`True`**

http://tinyurl.com/glideinWMS/doc.prd/factory/custom_vars.html#frontend_vars

Example match section

```
<match match_expr='glidein["attrs"]["GLIDEIN_Site"] in job["DESIRED_Sites"].split(","))'>
  <factory query_expr="GLIDEIN_Site=!=UNDEFINED”>
    <match_attrs>
      <match_attr name="GLIDEIN_Site" type="string"/>
    </match_attrs>
  </factory>
  <job query_expr="DESIRED_Sites=!=UNDEFINED”>
    <match_attrs>
      <match_attr name="DESIRED_Sites" type="string"/>
    </match_attrs>
  </job>
</match>
…
<attrs>
 …
  <attr name="GLIDECLIENT_Group_Start"
        value='(stringListMember(GLIDEIN_Site,DESIRED_Sites,",")=?=True)'
        glidein_publish="False" job_publish="False" parameter="True" type="string"/>

</attrs>
```

Comments

Comments in the config file

- The frontend config is an XML file
  - So will need XML-comaptible comments
- Two ways:
  - Plain XML comments
  - Frontend supported

Plain XML comments

- Standard XML comments always an option
  <!-- ... -->
- However, they will not survive XML manipulation
  - And every reconfig rewrites the XML by default to populate defaults for ease of reading
  - Can be disabled, by modifying frontend_startup

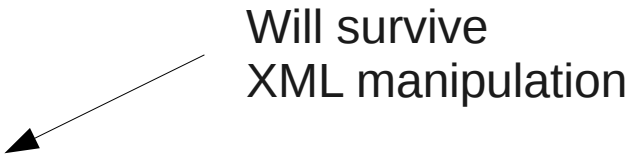
Not in RPM

```
<!-- Overflow only to select US sites -->
<factory
  query_expr='stringListMember(GLIDEIN_Site,"Nebraska,Wisconsin,UCSD,Purdue",",")'
  >

<!-- Don't need this anymore group name=”abc” -->
```

Frontend supported comments 1/2

- All XML tags in the Frontend config support the **`comment`** XML attribute
  - They are recognized by the parser
  - With the semantics of NOOP




```
<factory
    comment="Overflow only to select US sites"
    query_expr='stringListMember(GLIDEIN_Site,"Nebraska,Wisconsin,UCSD,Purdue",",")'
        >
```

Frontend supported comments 2/2

- Groups can also be disabled
  - Equivalent to commenting them out
- No equivalent for other XML tags, though

Actually, better.

Never remove a group as you will not be able to add it back!

```
<groups>
  <group name="Nebraska" enabled="True">
  …
  </group>
  <group name="Alaska" enabled="False">
  …
  </group>
<groups>
```

Putting it all together

- No way to put a complete XML file on a slide!
- Will do an interactive pass through

The End

Pointers

- The official glideinWMS project Web page is http://tinyurl.com/glideinWMS
- glideinWMS development team is reachable at glideinwms-support@fnal.gov
- The OSG glidein factory is reachable at osg-gfactory-support@physics.ucsd.edu

Acknowledgments

- The glideinWMS is a CMS-led project developed mostly at FNAL, with contributions from UCSD and ISI
- The glideinWMS factory operations at UCSD is sponsored by OSG
- The funding comes from NSF, DOE and the UC system