If the mountain won’t come to Mohammed... 
or: how to design a protocol for real world

Robert Schiele <rschiele@uni-mannheim.de>
FOSDEM 2006
The statements mentioned in this talk are solely based on my personal opinion and do not necessarily reflect the position of Novell, SUSE, or the openSUSE project.
What’s wrong with current protocol?
What’s wrong with current protocol?

- does not scale
What’s wrong with current protocol?

- does not scale
- not a problem of the protocol but of tool acceptance
What’s wrong with current protocol?

- does not scale
- not a problem of the protocol but of tool acceptance
- only one `drpmsync` server in public (mirror admins don’t like the tool)
What’s wrong with current protocol?

- does not scale
- not a problem of the protocol but of tool acceptance
- only one `drpmsync` server in public (mirror admins don’t like the tool)
- the only `drpmsync` server in public does also provide `rsync`, `ftp`, `http`, ... and is thus always overloaded
What’s wrong with current protocol?

- does not scale
- not a problem of the protocol but of tool acceptance
- only one `drpmsync` server in public (mirror admins don’t like the tool)
- the only `drpmsync` server in public does also provide `rsync`, `ftp`, `http`, ... and is thus always overloaded

⇒ protocol not ready for real world
How did I solve this?
How did I solve this?

- permanently annoying Christoph by whining about the badness of the world (did not help)
How did I solve this?

- permanently annoying Christoph by whining about the badness of the world (did not help)
- trying to teach `drpmsync` to fetch deltas from alternative servers (did not work because of lacking implementation documentation)
How did I solve this?

- permanently annoying Christoph by whining about the badness of the world (did not help)
- trying to teach `drpmsync` to fetch deltas from alternative servers (did not work because of lacking implementation documentation)
- implementing an alternative proof-of-concept implementation (`mydrpmsync`)
If the mountain won’t come to Mohammed...

mydrpmsync
mydrpmsync

- can use any rsync servers with mirrored deltas (scales with established mirror infrastructure)
mydrpmsync

- can use any `rsync` servers with mirrored deltas (scales with established mirror infrastructure)
- multithreaded (can be tuned not to suffer from a specific resource bottleneck)
mydrpmsync

- can use any \texttt{rsync} servers with mirrored deltas (scales with established mirror infrastructure)
- multithreaded (can be tuned not to suffer from a specific resource bottleneck)
- to be mirrored packages selectable (can mirror only required architecture packages)
If the mountain won’t come to Mohammed...

mydrpmsync

- can use any rsync servers with mirrored deltas (scales with established mirror infrastructure)
- multithreaded (can be tuned not to suffer from a specific resource bottleneck)
- to be mirrored packages selectable (can mirror only required architecture packages)
- robust (does neither crash nor do harm to your repository on failures like server inconsistencies or temporary transmission failures)
Are there alternative solutions?
Are there alternative solutions?

- implement a plugin infrastructure in `rsync` to handle specific files with specific algorithms
Are there alternative solutions?

- implement a plugin infrastructure in `rsync` to handle specific files with specific algorithms
- implement a `drpmsync` plugin for `rsync`
Are there alternative solutions?

- implement a plugin infrastructure in `rsync` to handle specific files with specific algorithms
- implement a `drpmsync` plugin for `rsync`
- convince mirror admins to use `rsync` with `drpmsync` plugin
How to improve the current protocol?
How to improve the current protocol?

- only distribute file metadata from `drpmsync` server
How to improve the current protocol?

• only distribute file metadata from `drpmsync` server
• fetch actual data with traditional protocols (`rsync`, `ftp`, `http`, ...) from other mirrors
How to improve the current protocol?

- only distribute file metadata from `drpmsync` server
- fetch actual data with traditional protocols (rsync, ftp, http, ...) from other mirrors
- solve inconsistencies between mirrors by a consistency controller (included in the drpmsync server)
questions, comments, rants...
If the mountain won’t come to Mohammed...

questions, comments, rants...

...should go to rschiele@uni-mannheim.de