



jabber inc.®

The Power of Presence™





A Mental Model for *Presence*

Simplicity as heresy

Joe Hildebrand
CTO
Jabber, Inc.



What's Presence?

- **Information about an identity that changes over time**
 - > Identity ≠ Person (groups, bots, apps, sensors, etc.)
 - > People are common, however. Design for them!
 - > If it's changing, publish/subscribe/notify is *required*
- **Changes come in chunks**
 - > Routability, capabilities of endpoint/device
 - > Geographic location
 - > Mood
- **Each chunk changes at a different rate**
- **Combination of chunks = full presence: *composition***



What's wrong?

- **Single document for all presence information**
 - > Server-side composition
 - > Composition ambiguity
 - > Filtering as access control
- **Transient subscriptions**
- **Querying/polling**
- **Bandwidth**
- **Infrastructure scale (memory, CPU)**
- **User expectation**
- **Complicated software on client and server**
- **Simultaneity is an illusion**

If there is more than one way to do it, interoperability is hard, at best.



What to do?

- **Define simple composition rules**
 - > Group by identity
 - > Last change wins (order matters)
- **Send minimal information**
 - > Initial sync, cache
 - > Updates
 - > Interest
- **Access control per chunk**
 - > Start with “do I know this person”
 - > Restrict further by user, group, domain, policy
 - > Only send updates to authorized watchers
- **Each watcher builds its own view**



The Power of Presence™



Why not?

- **RFC x says...**
 - > Standards are not chiseled into stone
 - > They were written by people
- **Composition is hard!**
 - > Only if it's done on the server
 - For each watcher: Compose, filter, diff, update
- **I only want to see *that guy's* mood**
 - > Register interest in mood, ignore for everyone but *that guy*
 - > You win overall in bandwidth, complexity goes way down
 - > By the way, your users rarely want this



The Power of Presence™

