Normalized Scopes for SLP

Charles E. Perkins

cperkins@eng.sun.com

http://www.srvloc.org/~charliep

Scope Design Points

- Scopes are sets of *services*
- Users typically select services by *attribute*, not by *scope*
- Users can be assigned scopes by the administration in order to conveniently segregate expected access patterns.
- Not an *access control* mechanism
- *local* scope defines a set of services that depends upon the user, and needs interpretation by the DA
- initial deployments won't use scopes (services are *unscoped*)
- smooth transitions into scoped administration – user agents that get scopes can still use existing unscoped services

Scope Handling Heuristics

For the sake of illustration, imagine that a service is a candy M&M, and that scopes divide the candy up into sets all containing the same color. Users want candy, but some users are more particular than others, and some users are constrained by their administration to only eat candy of a particular color.

User Agent Heuristics

tuples are <'S', scope-list>

<on, “”>	I want colorless M&Ms.
<on, “green”>	I want green M&Ms.
<off, “green”>	I want green M&Ms, or else colorless M&Ms.
<off, “”>	I want colorless M&Ms.

Service Agent Heuristics

<on, “”>	I am a colorless M&M.
<on, “green”>	I am a green M&M, and always appear colorful.
<off, “green”>	I am a green M&M, but I'm happy to appear colorless.
<off, “”>	I am a colorless M&M.

Directory Agent Heuristics

<on, “”>	I am a jar for colorless M&Ms.
<on, “green”>	I am a jar for green M&Ms only.
<off, “green”>	I am a jar for green M&Ms, but I’m happy to accept colorless.
<off, “”>	I am a jar for colorless M&Ms.