Restricting access to domain service endpoints

URL Pattern

(Ryan:) I have through around this a bit and I have been thinking of something even simpler. Just proxying requests with a specific URL pattern (say /api /providerregistry/*) and sending them to a specific configured domain service (so in this example the provider registry). So basically just acting as a proxy to the domain services.

More advanced options

(Carl:) While URL pattern matching would certainly work for me, and while I don't have any particular preference, I do think it is worthwhile to look at existing standards for making all these services known and to keep the software components as "pluggable" as possible.
I know WSDL has a bit extra overhead, but it may be worthwhile as consideration. Perhaps it's time for a spreadsheet to compare the pros/cons of different options e.g:

- URL pattern matching
- WSDL 1.1
- WSDL 2.0
- WADL
- any others?

against something like:

- existing libraries (Java, Python, PHP, etc)
- related security models/access control mechanisms
- supported "transport" mechanisms
  - proxy/pass-through
  - support for REST verbs
- compatibility with proposed inter-operabiltiy layer architecture
- compatibility with other IHE profiles
- etc.