Signing Proxy for Object Store

Signing proxy for object store provides REST interface, should be transparent for applications. Hosted close to end user providing extra integrity guarantees.

Signing
When a new object is pushed, its content is hashed on the fly (decoding content, joining multipart etc) and the hash signed using KSI. Signature token is stored (details TBD), link between object and signature token must be retained (details TBD).

Verification
Data is hashed on the fly during object retrieval, using exactly same decoding methods and hash algorithm (if hash algorithm ID is not extracted from signature token during signature verification, then it must be retained as part of the metadata somewhere). Associated signature token is retrieved, hash verified, result and signature properties returned.

Implementation
Use S3 REST API (more or less industry standard, http://docs.aws.amazon.com/AmazonS3/latest/API/) with OpenStack Swift (http://docs.openstack.org/developer/swift/) as the backend object store.

Preferred implementation language is Java. Whatever frameworks, application servers. Akka is interesting though.

Questions about design
- Where to store signatures? A database, object in the same object store, headers / metadata? Security considerations?
- How to return verification results (metadata)? HTTP headers?
- Signature can be generated only after entire object has been streamed through the proxy (hashed fully). Same for verification.
- Placement of this proxy? Security consideration?
- How is this proxy implementation different from S3 MD5 ETag? In terms of security guarantees?
- What to do if signature is "broken"?

Extra points for
- Custom metadata
- High-availability interface to KSI service
Necessary technical resources

- AWS (trial) account
- KSI service access
- KSI Java SDK

NDA is required to get access to the KSI service and development tools.

Contact
Risto Laanoja, risto.laanoja@guardtime.com