

Runtime Infrastructure - Bug #11987

Implement a SAN list for the certificate tools

10/13/2026 12:45 PM - Sergey Ivanovskiy

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		reviewer:	Constantin Asofiei
billable:	No	production:	No
vendor_id:	GCD	env_name:	
case_num:		topics:	
version_reported:			
version_resolved:			
Description			
Related issues:			
Related to Documentation - Support #9709: Needed: Basic example of configurin...			Test

History

- #2 - 10/13/2026 12:46 PM - Sergey Ivanovskiy
- Related to Support #9709: Needed: Basic example of configuring separate client and server added
- #3 - 10/13/2026 12:49 PM - Sergey Ivanovskiy
- File 11987-cert-san-task-spec.md added

FWD web certificate does not cover the hosts browsers connect to (remote broker clients, signature simulator)

Spun off from task 11409a (SIGNATURE widget / SigWeb simulator). Task: 11987. Background notes: 11409a/.tmp/11409a-simulator-cert-san.md.

Problem

An FWD server has **one web certificate**, and its SAN holds **one DNS name**. Every embedded HTTPS server that FWD starts presents that same certificate, but browsers reach those servers under **different host names**. When the name in the URL is not in the certificate's SAN, the browser rejects the connection.

Two known cases:

Case	URL the browser opens	Name the certificate must cover
Web client spawned by a remote broker	https://<broker remote:agent:host>:<client port>/index.html?<token>	each broker host's remote:agent:host (host name or IP address)
Signature pad simulator (11409a, webClient/signaturePadInSimMode)	https://tablet.sigwebtablet.com:47290/SigWeb/..	tablet.sigwebtablet.com

With the default FWD setup the certificate is also issued by a private project CA (e.g. hotel-root, valid until **2027-06-03**), so the browser first fails with ERR_CERT_AUTHORITY_INVALID. Once the CA is trusted, it fails with ERR_CERT_COMMON_NAME_INVALID. Users can click past the warning for a top-level page (remote broker client), but not for XHR calls (simulator, SigWebTablet.js). Even where clicking past works, Chrome stores the exception per host, so it has to be repeated for every broker host.

Root cause

One certificate, shipped to every client

- The web certificate is stored per server in the directory under /security/certificates/webCertificates/<serverId>/<alias,certificate,key/>...} (security/WebCertificates.java:85, getWebCertificates around :187). SecurityCache.java:636-662 loads it as the single webStore, under webServerAlias.
- ServerKeyStore.getStore() (main/ServerKeyStore.java:226-288) makes a copy for each spawn that contains the web key store, protected with fresh passwords. ServerKeyStore is Externalizable, so the web private key and certificate are serialized to the spawned client JVM, **including clients on remote broker hosts**.
- WebClientSpawner.java:203 → TemporaryClientTask → EmbeddedWebServer.startupServer(keyStore, ...) (:528). GenericWebServer.getSslContextFactory() (web/GenericWebServer.java:983-990) uses getWebKeyStore() when it is valid.
- SignatureSimulator.start() passes ServerKeyStore.getStore() to its own GenericWebServer, so the simulator presents the same web certificate.
- Nothing is keyed by broker host: the broker, spawner and key store code has no per-host alias, key store or directory node.

Remote clients are reached under the broker's name

- BrokerCore.initialize() (main/BrokerCore.java:456-476) sets dedicatedHost from remote:agent:host. The default is InetAddress.getLocalHost().getHostName() of the broker machine. The value may be a host name **or an IP address** (validated by Utils.isValidHostNameOrIpAddress).
- For a remote spawn, WebClientBuilder.java:173-177 passes client:web:host=\${remote.agent.host}. Also, WebClientSpawner.java:747-756 deliberately does **not** forward the configured webClient/host.
- EmbeddedWebServerImpl.getServerURI() (ui/client/driver/web/EmbeddedWebServerImpl.java:327-336) builds https://<host>:<port>/ from that value, and the browser is redirected there.
- Exceptions:
 - Dedicated mode forces the host to localhost (WebClientSpawner.java:513-520).
 - With webClient/portsRange/forwardedHost set, the browser sees only the reverse proxy name (WebClientSpawner.java:531-540).

The tooling can only write one name

- SSLCertGenUtil.generateCertificate(alias) (security/SSLCertGenUtil.java:1344-1365) takes the CN from the single string at /security/certificates/company/hostnames/<alias> (default "<alias> server|account").
- BCCertFactory.java:876 copies that CN into **one** dNSName SAN entry:

```
GeneralNames ext4 = new GeneralNames(new GeneralName(GeneralName.dNSName, cn));
```

It writes no further DNS names, no IP-address SANs and no wildcards.

- The only existing way around this is the admin REST upload (admin/server/CertRestHandler.java:189-224). It stores an externally issued PEM certificate/key/chain as the server's web certificate and hot-reloads it. That allows a cert with several SANs or a wildcard, but it is still one cert per server, made outside FWD.

Already done in 11409a (not part of this task)

- SignatureSimulator.warnIfHostNotCovered() logs a startup WARNING when no key entry in the simulator's key store lists tablet.sigwebtablet.com in its SAN. The check is SignatureSimulator.sanCoversHost() (DNS entries only, case-insensitive, one-label wildcard), covered by unit test SignatureSimulatorSanTest.

Scope

In scope

1. **Multi-name certificates from FWD tooling.** gencert.sh / SSLCertGenUtil can produce a certificate whose SAN lists several DNS names, IP addresses and *. wildcards.
2. **Remote broker clients.** A remote-broker web client presents a certificate that covers the host name the browser is redirected to, whether remote:agent:host is a host name or an IP address.
3. **Signature simulator.** The simulator presents a certificate that covers tablet.sigwebtablet.com.
4. **Diagnostics.** When an FWD embedded web server starts on a host name its certificate does not cover, a WARNING is logged (remote broker client start-up, alongside the existing simulator warning). The SAN matcher is shared by both, not duplicated.
5. **Documentation.** How to configure the names, regenerate the certificate, distribute it, and trust the project root CA (Chrome/NSS, Firefox, Windows). How to choose between multi-name certs, wildcards and a reverse proxy for remote brokers.

Out of scope

- Any change to SigWebTablet.js or p2j.signature.html. These client resources are frozen; the vendor SDK must stay pristine.
- Replacing the per-project private CA, or using a publicly trusted certificate (still possible through the admin upload).
- TLS between FWD client and server on the FWD protocol. Its SAN/CN check (SecurityManager.java:1258-1280, fetchAltNames) is not the browser-facing certificate.
- Renewing hotel-root before 2027-06-03. Track that separately; it is only noted here.

Options

Option	Change	Pros	Cons
--------	--------	------	------

A. Multi-name SAN in the generator (chosen)	/security/certificates/company/host names/<alias> takes a comma-separated list. The first entry becomes the CN, and every entry becomes a SAN entry: dNSName for host names and wildcards, IPAddress for IPv4/IPv6 literals. Regenerate with ./gencert.sh <alias>.	Small, generic change in one place. One cert covers the server, every broker host and tablet.sigwebtablet.com. Works with the existing "one web cert per server" distribution. No runtime change.	A new broker host means the certificate must be regenerated and the CA-signed cert redeployed. The list grows with the broker fleet (a wildcard keeps it short).
B. Per-broker-host certificate (possible follow-up)	Key a web certificate by remote:agent:host. When the spawn is remote, ServerKeyStore.getStore() / WebClientSpawner picks the broker's cert, falling back to the server web cert.	Each host gets a cert that covers exactly its own name. A broker's private key is not shared with other hosts.	New directory layout, plus admin UI/REST and generation support. More code in the security-sensitive path. Doesn't help the simulator by itself.
C. Wildcard only	*.example.com in the SAN (needs A's wildcard support, or an admin upload)	One short entry covers every broker in the domain	Doesn't cover IP addresses, localhost, or tablet.sigwebtablet.com, so in practice it is used together with A.
D. Reverse proxy (existing)	Set webClient/portsRange/forwardedHost, and terminate TLS on the proxy	No FWD change. The browser sees only one name.	Infrastructure to operate. Doesn't cover the simulator, which the browser reaches directly.

Decision (2026-10-13): option A, with the shared diagnostic (scope item 4). It removes the root limitation where it lives and covers both known cases with the existing distribution model. Options B–D are not part of this task. C (wildcards) and D (reverse proxy) remain available to deployments: C is supported by A's syntax, and D needs no FWD change. B is a separate follow-up, to be done only if a deployment requires per-host keys.

Requirements

- **R1.** The host name value accepts a list separated by commas. Whitespace around items and empty items are ignored.
- **R2.** The first entry is the subject CN. **All** entries become SAN entries, in configured order with duplicates removed: IPAddress for IPv4/IPv6 literals, and dNSName for everything else (including *. wildcards).
- **R3.** A single-name value produces exactly the same certificate as today (CN = SAN = that name).
- **R4.** Invalid input is rejected with a clear error: an empty list, an entry that is not a valid DNS name/wildcard/IP literal, or a wildcard that is not just the leftmost label. The tool must not produce a certificate without a SAN.
- **R5.** Move the SAN matcher (SignatureSimulator.sanCoversHost) into a shared security/web utility. Extend it to IPAddress entries, which must be compared as addresses, not as strings.
- **R6.** Remote-broker web client: after startupServer, if the URI host (not a forwardedHost URI, and not dedicated-mode localhost) is not covered by the certificate's SAN, log one WARNING. Name the host and the SAN entries, and point to the configuration that fixes it.
- **R7.** Signature simulator: keep the existing warning, now based on the shared matcher. No other simulator change is needed.
- **R8.** Docs (wiki / gencert.sh notes): the new syntax and examples, e.g. `fwdsrv.example.com,broker1.example.com,10.0.0.15,tablet.sigwebtablet.com`, with a wildcard variant. Explain how to trust the root CA, and when to use a proxy instead.

Acceptance criteria

1. **Generation.** Set the web alias host name list to `localhost,broker1.test,10.0.0.15,*.brokers.test,tablet.sigwebtablet.com` and run `./gencert.sh <alias>`. Then `openssl x509 -noout -ext subjectAltName` shows exactly those entries, with the IP as IP Address:.
2. **Single name, no regression.** Regenerating an alias that has one host name gives the same CN/SAN as before.
3. **Remote broker.**
 - Configure a broker whose `remote:agent:host` is in the list and the root CA is trusted. The browser opens the remote web client with no certificate warning, and no WARNING is logged.
 - Configure a broker whose host is **not** in the list. The client log shows the R6 WARNING.
 - Repeat both runs with `remote:agent:host` set to an IP address.
4. **Simulator.** With `tablet.sigwebtablet.com` in the list, `?simMode` loads with no certificate error and the "does not cover" warning is not logged. `IsSigWebInstalled()` succeeds and a signature is captured end to end. `openssl s_client -connect 127.0.0.1:47290 -servername tablet.sigwebtablet.com` shows the new cert.
5. **Unit tests** for R1–R5: single name, several names, whitespace, duplicates, wildcard, IPv4, IPv6, invalid input, and IP matching.

Open questions

- Could a directory `cfgOverrides / webOverrides` entry (appended later in `WebClientSpawner.java:776-784`) replace the broker-supplied `client:web:host` on the command line? If so, R6 must check the effective host. Override precedence in `BootstrapConfig.processOverrides` has not been traced.
- Does the server's main web server (`WebServer`, including `hotReloadSslCertificate`) use the same `webServerAlias` cert? This is assumed but not verified. It affects whether the server's own host name must stay in the list (it should anyway).

Risks / notes

- Certificate tooling is security-sensitive, so review it carefully. The CA key handling must not change.
- The web private key is already shared with every remote broker host (serialized in ServerKeyStore). Option A doesn't change that; option B would.
- Existing deployments keep working until they regenerate certificates; nothing changes automatically.
- hotel-root expires on 2027-06-03, after which every certificate it issued stops being valid.
- Real Topaz SigWeb takes the same approach (local HTTPS service, a cert for tablet.sigwebtablet.com, and the issuer installed into the OS trust store), so the CA trust step is expected for the simulator.

Files

11987-cert-san-task-spec.md	12.5 KB	10/13/2026	Sergey Ivanovskiy
-----------------------------	---------	------------	-------------------