

Codebase Analysis: API Server & CLI Client

Generated: 2026-04-07

Scope: apiserver/ (Go/Gin REST API) and xcecli/ (Go/Cobra CLI client)

Backend integration (scheduler daemon, C library) is out of scope.

1. System Architecture Overview

```
graph TD
    subgraph xcecli ["xcecli (CLI Client)"]
        CLI_CMD["cobra commands\njobs / hosts / queues\nroles / users / resources"]
        CLI_REQ["common/request\nAPIClient (resty)"]
        CLI_CFG["config\n~/ .xce/config.json\ntoken + server URL"]
        CLI_CMD --> CLI_REQ
        CLI_REQ --> CLI_CFG
    end

    subgraph apiserver ["apiserver (REST API Server)"]
        MW["middleware\nIP whitelist · URL whitelist\nJWT validate · rate limit"]
        ROUTER["router\nGin routes /xce/v1/..."]
        CTRL["controllers\njobs · hosts · queues\nroles · users · resources\nfiles · auth"]
        STORE["in-memory store\nDataStore[K,V] (sync.RWMutex)"]
        BACKEND["backend stub\nSubmitJob / ControlJob\nControlQueue / ControlHost"]
        AUTH_PROV["auth providers\nPAM · config-file · LDAP · OAuth2"]
        RBAC["RBAC engine\nroles · permissions · domains"]

        MW --> ROUTER
        ROUTER --> CTRL
        CTRL --> STORE
        CTRL --> BACKEND
        CTRL --> RBAC
        CTRL --> AUTH_PROV
    end

    subgraph future ["Backend (future integration)"]
        SCHED["Scheduler Daemon\n(C library / gRPC)"]
    end
```

```

CLI_REQ -- "HTTPS/HTTP\nBearer JWT" --> MW
BACKEND -. "TODO: C lib call" .-> SCHED
SCHED -. "notifications\nOnJobCreated/Updated" .-> BACKEND

```

2. API Server Completeness

2.1 Implemented Endpoints

Group	Method	Path	Permission	Status
Auth	POST	/xce/v1/auth/login	public	
Auth	POST	/xce/v1/auth/logout	public	
Permissions	GET	/xce/v1/permissions	rbac:view	
Roles	GET	/xce/v1/roles	rbac:view	
Roles	GET	/xce/v1/roles/:rolename	rbac:view	
Roles	POST	/xce/v1/roles/create	rbac:control	
Roles	POST	/xce/v1/roles/:rolename/createPermissions	rbac:control	
Roles	DELETE	/xce/v1/roles/:rolename/delete	rbac:control	
Users	GET	/xce/v1/users	rbac:view	
Users	GET	/xce/v1/users/:username	rbac:view	
Users	GET	/xce/v1/users/:username/permissions	rbac:view	
Users	POST	/xce/v1/users/:username/createRoles	rbac:control	
Users	DELETE	/xce/v1/users/:username/deleteRole	rbac:control	
Clusters	GET	/xce/v1/clusters	cluster:view	
Hosts	GET	/xce/v1/hosts	host:view	
Hosts	GET	/xce/v1/hosts/:hostname	host:view	
Hosts	POST	/xce/v1/hosts/select	host:view	
Hosts	POST	/xce/v1/hosts/control	host:control	
Queues	GET	/xce/v1/queues	queue:view	
Queues	GET	/xce/v1/queues/:queueName	queue:view	
Queues	POST	/xce/v1/queues/control	queue:control	
Jobs	GET	/xce/v1/jobs	job:view	
Jobs	GET	/xce/v1/jobs/:jobId	job:view	
Jobs	POST	/xce/v1/jobs/create	queue:submit	
Jobs	POST	/xce/v1/jobs/:jobId/control	job:control	
Jobs	POST	/xce/v1/jobs/config	job:control	
Jobs	POST	/xce/v1/jobs/:jobId/resubmit	job:control	
Resources	GET	/xce/v1/resources	resource:view	
Resources	GET	/xce/v1/resources/definition	resource:view	

Group	Method	Path	Permission	Status
Files	POST	/xce/v1/files/upload	file:control	
Files	GET	/xce/v1/files/:uid/:filename	file:view	
Files	GET	/xce/v1/files	file:view	

2.2 Missing / Incomplete Areas

- **File DELETE** — route is commented out (`//filesApi.DELETE(...)`) but listed in the URL whitelist. No implementation.
- **CORS** — middleware is commented out. Cross-origin requests from web UIs will fail.
- **Swagger docs are stale** — `docs/swagger.yaml` still references old `models.BasicHost` / `models.FullHost` types and `/api/v1/hosts`, which no longer match the actual routes (`/xce/v1/hosts`) or current models.
- **Backend stubs** — all `backend/stub.go` functions (`SubmitJob`, `ControlJob`, `ControlQueue`, `ControlHost`) return `nil` with a `// TODO` comment. The server accepts requests but does not actually dispatch them to the scheduler.
- **Notification path** — `backend/handler.go` defines `OnJobCreated/Updated/Deleted` etc., but nothing calls these handlers yet. The in-memory store is never populated from the real backend.
- **API key management** — `models/apikey.go` and `controllers/apikey.go` exist but no route is registered for API key CRUD.
- **No pagination on control responses** — control endpoints return `accepted/failed` lists but no page token.

3. CLI Client Completeness

3.1 Implemented Commands

Command	Sub-commands	Coverage
<code>xce login</code>	—	
<code>xce logout</code>	—	
<code>xce list</code>	(auth list)	
<code>xce jobs</code>	list, submit, control, restart, files	
<code>xce hosts</code>	list, control	
<code>xce queues</code>	list, control	
<code>xce clusters</code>	list	
<code>xce resources</code>	list (dist + def)	

Command	Sub-commands	Coverage
xce roles	list, view, add, delete	
xce users	list, view, assign-role, unassign-role, perm4user	
xce permissions	list	

3.2 Missing CLI Coverage

- **File delete** — no `xce files delete` command (matches missing server endpoint).
- **Job array stats** — server supports `?type=array_stats` but no dedicated CLI flag/display path for it.
- **API key management** — no CLI commands for creating/listing/revoking API keys.
- **Cluster detail** — only `list`; no per-cluster detail view.
- **TLS cert pinning** — `SetRootCAs` always sets `InsecureSkipVerify: true` regardless of the cert path passed in; the `loadRootCAs` helper is never actually used for the logon path.

4. Client–Server Communication Gaps

```
sequenceDiagram
    participant CLI as xcecli
    participant SRV as apiserver
    participant STORE as in-memory store
    participant BE as backend stub

    CLI->>SRV: POST /xce/v1/auth/logon {username, password}
    SRV-->>CLI: {token, token_expire_time}
    Note over CLI: saves token to ~/.xce/config.json

    CLI->>SRV: POST /xce/v1/jobs/create Bearer <token>
    SRV->>BE: SubmitJob() [stub - returns nil]
    BE-->>SRV: nil (no real dispatch)
    SRV-->>CLI: {jobid, queue_name}
    Note over STORE: store NOT updated until OnJobCreated fires (never)

    CLI->>SRV: GET /xce/v1/jobs Bearer <token>
    SRV->>STORE: query in-memory store
```

```
STORE-->>SRV: [] (empty - backend never populated it)
SRV-->>CLI: {count:0, rows:[]}
```

4.1 Identified Gaps

#	Area	Server side	Client side	Impact
G1	Token format	Sends <code>token</code> string in logon response	Stores and sends as <code>"Bearer <token>"</code> with a leading space (<code>" " + token</code>)	Server strips <code>"Bearer "</code> prefix correctly, but the extra space in <code>SubmitJob (" "+token)</code> vs no space in <code>GetRequest (" "+token)</code> is inconsistent and may cause auth failures on some endpoints
G2	JobSubmitRequest.JobName required	<code>JobName</code> required	<code>binding:"required"</code>	Client enforces job name; server does not — mismatch in validation semantics
G3	JobControlRequest.Method field (e.g. <code>"criu"</code>)	<code>Method</code> field (e.g. <code>"criu"</code>)	<code>JobControlJobIDRequest</code> has no <code>method</code> field	Checkpoint method cannot be specified from CLI
G4	HostInfo.NumCPUs field <code>num_cpus</code> (int)	<code>num_cpus</code> (int)	Mapped to <code>num_cores</code> in <code>XCEHost</code>	Wrong field name mapping — CPU count will always be 0 on client display

#	Area	Server side	Client side	Impact
G5	QueueInfo.QueueStatus	<code>status</code> (bitmask)	<code>string</code> in <code>DBQueueInfo</code>	Server sends integer; client expects string — JSON unmarshal will fail silently or error
G6	JobInfo.ExitStatus	<code>status</code> (raw JSON string)	<code>int</code> in <code>QueryJob</code>	Type mismatch — client will fail to parse exit status
G7	ResourceDist value field	<code>numeric_value</code> / <code>boolean_value</code> / <code>string_value</code> (separate fields)	<code>ResourceDist.ResourceValue</code> mapped from <code>"value"</code>	Server values typed fields; client expects a single <code>"value"</code> key — will always be empty
G8	Token expiry check	Server validates JWT signature + expiry on every request	Client checks <code>createTime + tokenExpireTime</code> locally	Clock skew or server-side token revocation (logout) not reflected in client state
G9	TLS	HTTPS optional, self-signed cert included	<code>InsecureSkipVerify</code> <code>true</code> hardcoded for logon; cert path accepted but <code>loadRootCAs</code> never wired in	No real TLS verification in practice
G10	Swagger base path	Actual routes under <code>/xce/v1/...</code>	—	Swagger doc says <code>host: localhost:8443</code> and <code>/api/v1/hosts</code> — completely out of sync

5. Potential Risks

5.1 Security

Risk	Severity	Detail
Hardcoded JWT secret	High	<code>secretKey: "xce-api-key"</code> in <code>config/app.yaml</code> committed to repo. Anyone with the key can forge tokens.
Token stored as plain file	Medium	Server writes JWT to <code>tokens/<username>/token</code> on disk. File permissions are 0600 but the token is long-lived (86400 s default) and not invalidated on server restart.
IP binding in JWT	Medium	Token is bound to client IP at login. NAT/proxy environments or IP changes will silently reject valid users.
TLS InsecureSkipVerify	Medium	Client disables cert verification for logon. MITM attack can steal credentials.
Mock RBAC data	Medium	<code>rbac_check.go</code> contains hardcoded <code>hostGroupMap</code> , <code>userGroupMap</code> , <code>queueMap</code> used in domain checks — these are test fixtures left in production code.
<code>ioutil.ReadFile</code> (deprecated)	Low	<code>rbac_check.go</code> uses <code>ioutil.ReadFile</code> ; minor but signals code age.
Rate limiter per-endpoint map not cleared	Low	<code>limiters</code> map in <code>rate_limit.go</code> grows unboundedly as new route patterns are seen.

5.2 Reliability

Risk	Severity	Detail
Entire state is in-memory	High	All jobs, hosts, queues, roles are in <code>DataStore</code> (RAM). Server restart loses all state. No persistence layer.
Backend stubs always succeed	High	<code>SubmitJob()</code> returns <code>nil</code> — the server tells the client the job was submitted even though nothing happened.
No backend notification wiring	High	<code>DefaultHandler</code> methods exist but are never registered/called. Store is never populated from real data.
Snowflake ID clock regression	Medium	<code>jobidgen.go</code> returns an error on clock regression but callers in <code>controllers/jobs.go</code> may not handle it gracefully.
Single cluster ID hardcoded	Medium	<code>clusterID = 1</code> is a constant in <code>controllers/jobs.go</code> . Multi-cluster support is not wired.
No graceful shutdown	Low	<code>cmd/main.go</code> does not handle <code>SIGTERM</code> / <code>SIGINT</code> for in-flight request draining.

5.3 Correctness

Risk	Severity	Detail
Type mismatches (G4–G7 above)	High	Several model fields differ between server and client — silent data loss or parse errors.

Risk	Severity	Detail
Whitelist pattern matching bug	Medium	<code>WhitelistURLMiddleware</code> uses <code>pattern[:len(pattern)-1]</code> to strip trailing <code>*</code> , but patterns without <code>*</code> (e.g. <code>/xce/v1/files</code>) are also truncated by one character, potentially blocking valid paths.
<code>validateRequestBody</code> only for non-multipart POST	Low	Logic flag <code>:= !strings.Contains("multipart/form-data" ...)</code> is inverted — <code>strings.Contains</code> checks if the first arg contains the second, not the other way around. The condition is always <code>false</code> for multipart, which is correct by accident.

6. Code Quality Evaluation

6.1 Strengths

- **Clean layering** — router → middleware → controller → store/backend is well-separated.
- **Generic DataStore** — `DataStore[K,V]` with `sync.RWMutex` is a solid, reusable pattern.
- **Structured logging** — `go.uber.org/zap` used consistently with field-level context.
- **Config schema validation** — JSON schema files for `app.yaml`, `db.yaml`, `logger.yaml`.
- **Auth provider abstraction** — PAM / config-file / LDAP / OAuth2 behind a `Provider` interface.
- **Rate limiting** — per-endpoint token-bucket limiter with configurable window.
- **CLI auth middleware** — `withAuth` wrapper cleanly intercepts all commands requiring login.

6.2 Issues & Improvement Suggestions

```
mindmap
  root((Improvements))
    Security
      Rotate JWT secret via env var
      Remove hardcoded mock RBAC data
      Fix TLS cert verification in client
    Reliability
      Add persistence layer (PostgreSQL already in go.mod)
      Wire backend notification handler
      Graceful shutdown with context
    Correctness
      Align model types server client
      Fix whitelist pattern truncation
      Fix token header space inconsistency
    Maintainability
      Regenerate Swagger docs
      Replace ioutil with io/os
      Add integration tests
      Extract clusterID to config
    Observability
      Add request-ID middleware
      Expose /metrics (Prometheus)
      Structured access log
```

Area	Issue	Recommendation
Models	Server and client define duplicate structs with diverging types	Extract shared types into a common Go module or generate from OpenAPI spec
Swagger	Docs are stale and reference wrong paths/types	Run <code>swag init</code> after fixing annotations; add to CI
Error codes	<code>common.go</code> defines numeric codes but HTTP status codes are always 200 even for errors	Use proper HTTP status codes (400, 401, 403, 404, 429, 500)
Token header	Inconsistent " "+token vs ""+token in client	Standardize to "Bearer "+token everywhere

Area	Issue	Recommendation
Mock data	hostGroupMap, userGroupMap, queueMap in rbac_check.go	Remove; load from store or config
ioutil	Deprecated since Go 1.16	Replace with <code>os.ReadFile</code> / <code>io.ReadAll</code>
Test coverage	Only <code>utils_test.go</code> , <code>rbac_check_test.go</code> , <code>store_test.go</code> , <code>audit_test.go</code> , <code>config_file_test.go</code>	Add controller-level and integration tests
clusterID	Hardcoded const <code>clusterID = 1</code>	Move to app config
File delete	Commented-out route	Either implement or remove from whitelist

7. Third-Party Dependencies

7.1 API Server (apiserver/go.mod)

Package	Version	Purpose
<code>github.com/gin-gonic/gin</code>	v1.10.0	HTTP web framework
<code>github.com/golang-jwt/jwt/v4</code>	v4.5.1	JWT generation & validation
<code>github.com/google/uuid</code>	v1.6.0	UUID generation for file uploads
<code>github.com/lib/pq</code>	v1.10.9	PostgreSQL driver (in go.mod but store is in-memory)
<code>github.com/msteinert/pam</code>	v1.2.0	PAM authentication (Linux only)
<code>github.com/spf13/viper</code>	v1.19.0	Configuration file loading
<code>github.com/swaggo/gin-swagger</code>	v1.6.0	Swagger UI integration
<code>github.com/swaggo/swag</code>	v1.16.4	Swagger doc generation
<code>github.com/xeipuuv/gojsonschema</code>	v1.2.0	JSON schema validation for config
<code>go.uber.org/zap</code>	v1.27.0	Structured logging
<code>golang.org/x/time</code>	v0.9.0	Token-bucket rate limiter

Package	Version	Purpose
<code>gopkg.in/natefinch/lumberjack</code>	<code>v2.2.1</code>	Log rotation
<code>gopkg.in/yaml.v2</code>	<code>v2.4.0</code>	YAML config parsing
<code>github.com/AlecAivazis/survey</code>	<code>v2.2.1</code>	Interactive prompts (unused in server — likely leftover)

7.2 CLI Client (`xcecli/go.mod`)

Package	Version	Purpose
<code>github.com/spf13/cobra</code>	<code>v1.9.1</code>	CLI command framework
<code>github.com/spf13/viper</code>	<code>v1.20.1</code>	Configuration loading
<code>github.com/xeipuuv/gojsonschema</code>	<code>v1.2.0</code>	JSON schema validation
<code>go.uber.org/zap</code>	<code>v1.27.0</code>	Structured logging
<code>golang.org/x/term</code>	<code>v0.27.0</code>	Terminal password input
<code>gopkg.in/natefinch/lumberjack</code>	<code>v2.2.1</code>	Log rotation
<code>gopkg.in/yaml.v2</code>	<code>v2.4.0</code>	YAML config parsing
<code>resty.dev/v3</code>	<code>v3.0.0-beta.2</code>	HTTP client (beta — not production stable)

`resty.dev/v3` is a **beta** release. Consider pinning to a stable `v2` (`github.com/go-resty/resty/v2`) for production use.

8. Recommended Next Steps (Priority Order)

```
gantt
    title Recommended Work Items
    dateFormat YYYY-MM-DD
    section Critical
    Fix model type mismatches (G4-G7)           :crit, a1, 2026-04-08, 5d
    Wire backend notification handler             :crit, a2, 2026-04-08, 7d
    Remove hardcoded JWT secret                  :crit, a3, 2026-04-08, 2d
    section High
    Fix token header space inconsistency         :b1, after a3, 2d
    Fix whitelist pattern truncation bug         :b2, after a3, 1d
    Remove mock RBAC data                       :b3, after a1, 3d
    Regenerate Swagger docs                     :b4, after a1, 2d
```

section Medium	
Add persistence (PostgreSQL)	:c1, after a2, 10d
Implement file delete endpoint	:c2, after b2, 2d
Fix TLS cert verification in client	:c3, after b1, 3d
Graceful shutdown	:c4, after a2, 2d
section Low	
Replace ioutil with os/io	:d1, 2026-04-15, 1d
Add integration tests	:d2, 2026-04-15, 7d
Upgrade resty to stable v2	:d3, 2026-04-15, 1d
Add Prometheus metrics endpoint	:d4, 2026-04-20, 3d