



EVESEAT

ROADMAP

A proposal of primary development effort and direction

Section 1

User Survey Results

1.1 Abstract

SeAT is primarily installed by corporations, followed by alliances. Few installations are for smaller groups. Within the ranked usage patterns, SeAT seems to be most deployed for Management and Admin activities of the group, including recruit screening and counterintelligence operations.

Development efforts should focus on improving backend performance / job throughput and resource usage, followed by working to get down the backlog of reported issues. Frontend performance is not a high priority. The community is happy with older versions of packages so long as vulnerabilities are patched, though there is also a desire to keep up to date.

In the community, there are more plugin developers than anticipated. However, it is important to note that a majority of plugin developers are not publishing their plugins. There is strong support for establishing a centralised organisation for plugins. Though inclusion of plugins is mixed as to whether to automatically include inactive plugins or be purely opt-in.

1.2 Aggregate Results

For context, there were over 60 responses to the survey. Percentages have been rounded to nearest whole number and made to total 100%. In the case of 'Other' / custom responses I have interpreted the answer and applied the percentage to the closest answers (including split where applicable).

Organisation Type	Percentage
Coalition	1
Alliance	20
Corporation	73
Small Group	3
Individual	3

Usage Pattern	Calculated Ranking Score
Corporation Management and Administration	240
Recruit Screening	230
Counter Intelligence (Spy Hunting)	213
Alt Character Management	202
Communications Role Management	146
Intelligence Gathering / Management (Espionage)	141
Industrial Management	116

SeAT 6 Upgrade Effort	Percentage Response
Happy to have some manual update work required	36
No Preference	32
Prefer to have manual update work similar (5.0.1 > 5.0.2)	27

Software Version Preference	Percentage Response
Older Versions OK with Patching	49
Must update to latest	30
Older Versions OK if vulns declared	21

Development Priority Preference	Calculated Ranking Score
Backend Job Performance	158
GitHub Issues	111
Preparations for SeAT v6	100
Frontend Performance	91

Are you a Plugin Developer	Percentage
No	82
Yes - Private Plugins	12
Yes - Publicly available plugin	6

Do you support a centralised plugin organisation	Percentage
I fully support this, including cloning inactive repos in	61
I support the organisation, opt-in by plugin developers.	31
I do not support the organisation	6

Section 2

Development Efforts

Four main streams of effort are proposed going forward with SeAT core development.

- Backend Performance Improvement
- Github Issue Backlog
- Plugin Organisation Standup
- SeAT 6

There is some intermingling of effort with SeAT 6, so a little bit of duplication may be present below. Also, a section on SeAT 5.1 has been added to consolidate those into one place; however, all entries are duplicates elsewhere.

The header of each card will detail the milestone the item is intended to fit into.

2.1 Backend Performance

Backend performance improvements involve an effort to decrease the runtime, CPU, RAM and IO usage.

SeAT v5.0: Cache Reintegration

In the move from v4 to v5 the cache usage of ‘eveapi’ was significantly reduced. This leads to ensuring that data is always attempted to be updated even if it has not changed. Due to the way Laravel works, if the data has not changed the insert is relatively lightweight to the DB, however it is still extra latency and IOPS that can potentially be avoided.

Therefore a reintroduction in appropriate locations should help improve job performance.

This is being addressed in SeAT v5.0.x (<https://github.com/eveseat/eveapi/pull/413>)

SeAT v5.1: Lite Cache

The current cache implementation stores the entire dataset related to the request. This data is not used in a large number of cases and as such is an extra overhead that can be avoided.

This involves work both within eseye and eveapi to make this possible.

eseye Provide the ability to configure multiple levels of caches

eseye Provide a method of allowing a call to ESI to be routed to an appropriate cache.

eveapi Provide an env configurable method to set up two cache tiers (normal and lite)

eveapi For each job assess and mark ability to use lite cache

SeAT v5.1: PHP 8.3

Version bump supported PHP version to 8.3 (docker image also bumped)

SeAT v5.1: Benchmarking

Develop a repeatable benchmark suite to assess relevant performance metrics on the entire stack. In particular initial focus will be on ESI Job execution. This is to aid in future development to be aware of degradation due to changes, as well as improvements for optimisations.

SeAT v6.0: Redis

Investigate redis forks for alternatives. Selection based on speed primarily.

SeAT v6.0: Postgres

This is dependant on DB agnostic operation (discussed in SeAT v6 stream)

Postgres has been trialled previously as it was a candidate inclusion for SeAT v5. It was not pursued due to time and compatibility with migrations requirements. One of the major hurdles is bringing data forward from MariaDB to Postgres. Perhaps an approach similar to [this](#).

2.2 Github Issue Backlog

SeAT v5.0: Triage

A triaging effort in order to assess all current open tickets. If the ticket is no longer valid, it will be closed. The remaining issues will be assessed for:

- Age
- Impact
- Complexity

SeAT v5.1: Burndown

Once triaged, issues will be worked on for burndown primarily targeting SeAT v5.1 however movement to 5.0 or 6.0 as required.

SeAT v5.1: Analytics

Correctly implemented analytics can also help guide how frequently certain errors appear within the wider community. The current analytics system is broken and in need of an overhaul. There are three main options; stay with Google Analytics and update to the current generation GA, find another free analytics solution, self-hosted analytics platform.

2.3 Plugin Organisation Standup

The Plugin Organisation will be created external to the eveseat organisation. This is to ensure a separation of concerns regarding repository permission access.

The intent for the plugin repository is that any plugins within the organisation will be managed primarily by their own plugin owners/authors. However, in order to help maintain plugins going forward there will be a leadership team established who will also be able to commit to each plugin in order to incorporate changes in circumstances when the plugin owner is not available to do so. Prime examples of when this may occur could be:

- Patching critical vulnerabilities
- Updating to new major SeAT versions while the plugin maintainer is unavailable

The primary intent of the organisation will be to have plugin developers opt-in to the organisation with their plugins. However, on a case-by-case basis plugins which are considered valuable enough to the community where the maintainers are inactive may be forked into the organisation. In the event this is to occur, an attempt to contact the original developers will be made before the fork. If the plugin is forked with no response then a new primary maintainer will be nominated for the fork with the ability for the original developer to be reincorporated if they return.

The plugin organisation will not operate as a free-for-all feature development location. Feature development will be done by the plugin owners themselves.

2.4 SeAT 6

SeAT v6.0: Laravel Update

Update to later release of Laravel. Probably 12 but will depend based on when this occurs.

SeAT v6.0: Redis

Investigate redis forks for alternatives. Selection based on speed primarily.

SeAT v6.0: BS5 / AdminLTE v4

AdminLTE v4 will hopefully be available fully by Laravel 12 release. Beta releases are already available

SeAT v5.1: BS5 / AdminLTE v4

In SeAT 5.1 we can start the transition over to BS5 / Admin LTE v4 by including both v3 and v4 of AdminLTE. This will allow for developing the new templates using the BETA releases. And allow users to OPT-IN to BS5 while this work is conducted. The main intent here is to allow plugin developers to work with BS5 before a hard switch is made in SeAT 6.0

SeAT v6.0: DB Agnostic Operations

THIS IS A MAJOR EFFORT.

The entire stack should be DB agnostic. The first benefit is that Postgres can be used as a more performant database for potential performance improvements. The second is that unit testing can utilise a full database migration within SQLite instead of the extract we currently have.

SeAT v6.0: PHP 8.4

Should be available when Laravel 12 is.

SeAT v6.0: Stack Monitoring

Possible thoughts here:

1. Prometheus / Grafana
2. <https://github.com/eveseat/services/pull/147>

2.5 SeAT 5.1

Entries here are duplicated from above sections.

SeAT v5.1: BS5 / AdminLTE v4

In SeAT 5.1 we can start the transition over to BS5 / Admin LTE v4 by including both v3 and v4 of AdminLTE. This will allow for developing the new templates using the BETA releases. And allow users to OPT-IN to BS5 while this work is conducted. The main intent here is to allow plugin developers to work with BS5 before a hard switch is made in SeAT 6.0

SeAT v5.1: Burndown

Once triaged, issues will be worked on for burndown primarily targeting SeAT v5.1 however movement to 5.0 or 6.0 as required.

SeAT v5.1: Analytics

Correctly implemented analytics can also help guide how frequently certain errors appear within the wider community. The current analytics system is broken and in need of an overhaul. There are three main options; stay with Google Analytics and update to the current generation GA, find another free analytics solution, self-hosted analytics platform.

SeAT v5.1: Lite Cache

The current cache implementation stores the entire dataset related to the request. This data is not used in a large number of cases and as such is an extra overhead that can be avoided.

This involves work both within eseye and eveapi to make this possible.

eseye Provide the ability to configure multiple levels of caches

eseye Provide a method of allowing a call to ESI to be routed to an appropriate cache.

eveapi Provide an env configurable method to set up two cache tiers (normal and lite)

eveapi For each job assess and mark ability to use lite cache

SeAT v5.1: PHP 8.3

Version bump supported PHP version to 8.3 (docker image also bumped)

SeAT v5.1: Benchmarking

Develop a repeatable benchmark suite to assess relevant performance metrics on the entire stack. In particular initial focus will be on ESI Job execution. This is to aid in future development to be aware of degradation due to changes, as well as improvements for optimisations.