

# External Dependencies for Java

## References

- [jqassistant](#)
- [Neo4j Python Driver](#)

## External Package Usage

### External Package

An external type has no `byteCodeVersion` since it only occurs as a dependency but wasn't analyzed itself (missing bytecode). Core Java types like `java.lang.Integer` and primitives like `int` are considered "build-in" and therefore aren't interpreted as "external" even though their byte code is also missing. A package is categorized as "external" if the types it contains are classified as external.

### External annotation dependency

The aforementioned classification encompasses external annotation dependencies as well. These dependencies introduce significantly less coupling and are not indispensable for compiling code. Without the external annotation the code would most probably behave differently. Hence, they are included in the first more overall and general tables and then left out in the later more specific ones.

### Table 1 - Top 20 most used external packages overall

This table shows the external packages that are used by the most different internal types overall. Additionally, it shows which types of the external package are actually used. External annotations are also listed.

Only the top 20 entries are shown. The whole table can be found in the following CSV report:

`External_package_usage_overall`

#### Columns:

- *externalPackageName* identifies the external package as described above

- *numberOfExternalCallerPackages* refers to the distinct packages that make use of the external package
- *numberOfExternalCallerTypes* refers to the distinct types that make use of the external package
- *numberOfExternalTypeCalls* includes every dependency to the types in the external package
- *numberOfExternalTypeCallsWeighted* includes every invocation or reference (sum of weights) to the types in the external package
- *allPackages* contains the total count of all analyzed packages in general
- *allTypes* contains the total count of all analyzed types in general
- *externalTypeNames* contains a list of actually utilized types of the external package

	externalPackageName	numberOfExternalCallerPackages	numberOfExternalCallerTypes	numberOfExternalTypeCalls	numberOfExternalTypeC
0	javax.annotation	76	342	374	
1	org.slf4j	66	141	253	
2	javax.persistence	15	27	82	
3	com.fasterxml.jackson.annotation	13	23	57	
4	jakarta.persistence	10	25	73	
5	io.axoniq.axonserver.grpc	7	30	55	
6	io.axoniq.axonserver.connector	6	21	30	
7	org.hamcrest	5	27	59	
8	io.grpc	4	12	50	
9	reactor.core.publisher	4	27	49	
10	com.github.kagkarlsson.scheduler	3	5	8	
11	com.github.kagkarlsson.scheduler.task	3	5	11	
12	io.grpc.stub	3	6	7	
13	org.jobrunr.scheduling	3	5	7	
14	org.springframework.boot.actuate.health	3	4	7	
15	org.springframework.boot.autoconfigure	3	24	54	
16	org.springframework.boot.autoconfigure.condition	3	32	66	
17	org.springframework.context.annotation	3	30	41	
18	com.fasterxml.jackson.databind	2	9	17	
19	com.thoughtworks.xstream.io	2	4	9	

Table 1 Chart 1a - Most called external packages in % by types (more than 0.7% overall)

External packages that are used less than 0.7% are grouped into the name "others" to get a cleaner chart with the most significant external packages and how often they are called in percent.

<Figure size 640x480 with 0 Axes>

Top external package usage [%] by type (more than 0.7% overall)

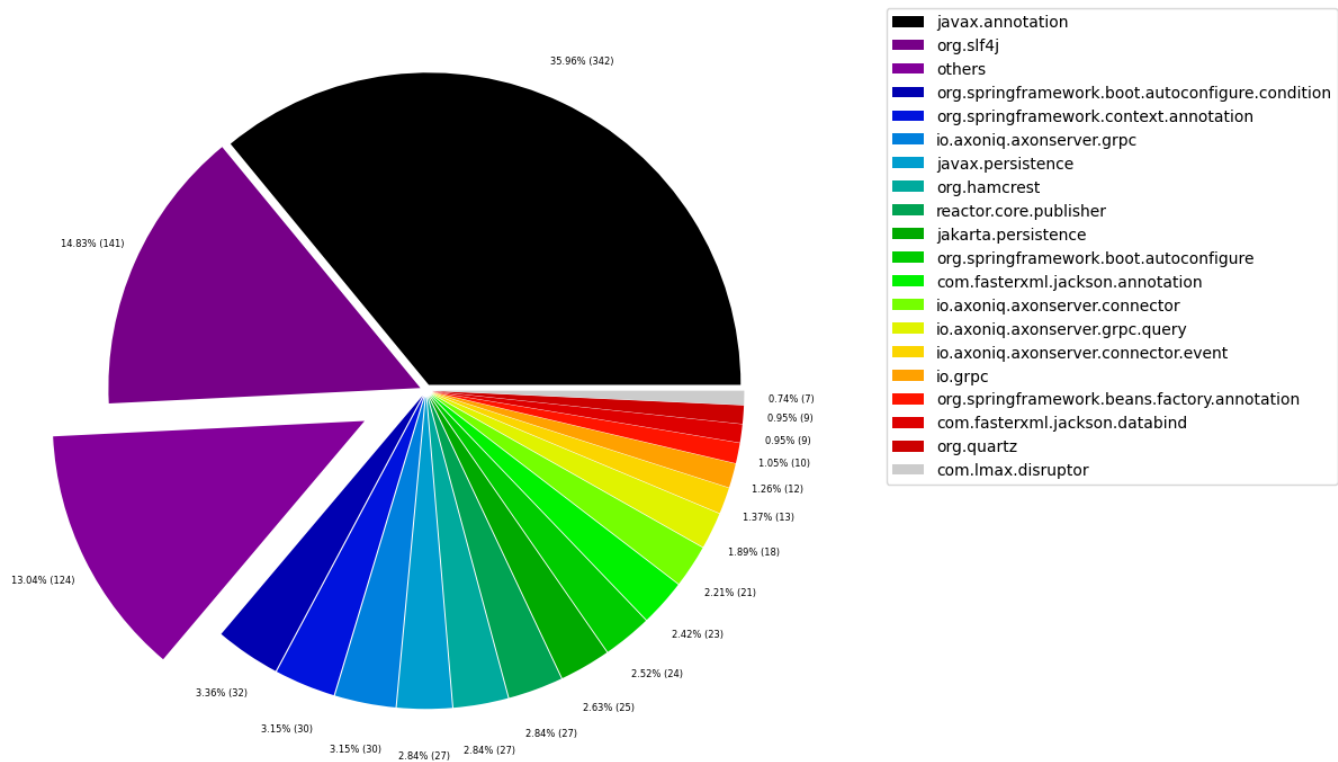


Table 1 Chart 1b - Most called external packages in % by types (less than 0.7% overall "others" drill-down)

Shows the lowest (less than 0.7% overall) most called external package. Therefore, this plot breaks down the "others" slice of the pie chart above. Values under 0.3% from that will be grouped into "others" to get a cleaner plot.

<Figure size 640x480 with 0 Axes>

Top external package usage [%] by type (less than 0.7% overall "others" drill-down)

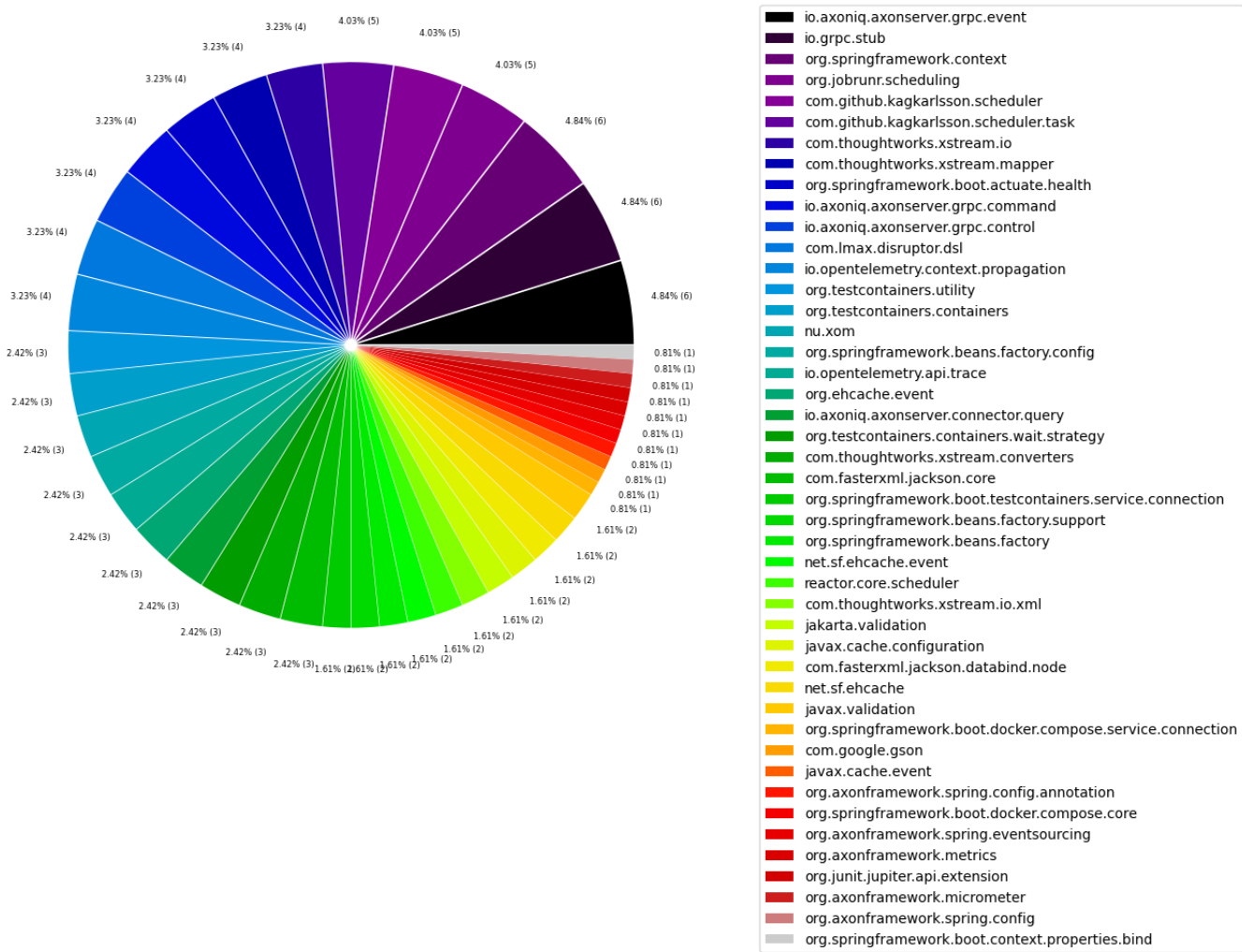


Table 1 Chart 2a - Most called external packages in % by packages (more than 0.7% overall)

External packages that are used less than 0.7% are grouped into the name "others" to get a cleaner chart with the most significant external packages and how often they are called in percent.

<Figure size 640x480 with 0 Axes>

Top external package usage [%] by package (more than 0.7% overall)



Table 1 Chart 2b - Most called external packages in % by packages (less than 0.7% overall "others" drill-down)

Shows the lowest (less than 0.7% overall) most called external package. Therefore, this plot breaks down the "others" slice of the pie chart above. Values under 0.3% from that will be grouped into "others" to get a cleaner plot.

<Figure size 640x480 with 0 Axes>

Top external package usage [%] by package (less than 0.7% overall "others" drill-down)



Table 2 - Top 20 most used external packages grouped by their first 2 layers

This table shows external packages grouped by their first 2 layers that are used by the most different internal types overall including external annotations. For example, "javax.xml.stream" and "javax.xml.parsers" are grouped together to "javax.xml".

Additionally, it shows which types of the external packages are actually used.

Only the top 20 entries are shown. The whole table can be found in the following CSV report:

External\_second\_level\_package\_usage\_overall

### Columns:

- externalSecondLevelPackageName* identifies the first 2 levels of the external package as described above
- numberOfExternalCallerPackages* refers to the distinct packages that make use of the external package
- numberOfExternalCallerTypes* refers to the distinct types that make use of the external package

- *numberOfExternalTypeCalls* includes every dependency to the types in the external package
- *numberOfExternalTypeCallsWeighted* includes every invocation or reference (sum of weights) to the types in the external package
- *allPackages* contains the total count of all analyzed packages in general
- *allTypes* contains the total count of all analyzed types in general
- *externalTypeNames* contains a list of actually utilized types of the external package

	externalSecondLevelPackageName	numberOfExternalCallerPackages	numberOfExternalCallerTypes	numberOfExternalTypeCalls	numberOfExternalTypeCallsWeighted
0	javax.annotation	76	342	374	17.0
1	org.slf4j	66	141	253	7.0
2	com.fasterxml	15	33	89	1.0
3	javax.persistence	15	27	82	3.0
4	jakarta.persistence	10	25	73	2.0
5	io.axoniq	9	67	197	9.0
6	org.springframework	8	53	249	5.0
7	io.grpc	5	18	61	1.0
8	org.hamcrest	5	27	59	3.0
9	reactor.core	5	28	53	2.0
10	com.google	4	9	12	0.0
11	com.github	3	7	21	1.0
12	com.thoughtworks	3	10	33	1.0
13	org.jobrunr	3	5	7	0.0
14	org.axonframework	2	7	17	0.0
15	org.junit	2	4	8	0.0
16	org.quartz	2	9	36	1.0
17	com.lmax	1	7	14	0.0
18	io.micrometer	1	1	2	0.0
19	io.opentelemetry	1	5	21	0.0

Table 2 Chart 1a - Most called second level external packages in % by type

External package groups that are used less than 0.7% are grouped into the name "others" to get a cleaner chart with the most significant external packages and how often they are called in percent.

<Figure size 640x480 with 0 Axes>

Top external package (grouped by first 2 layers) usage [%] by type (more than 0.7% overall)

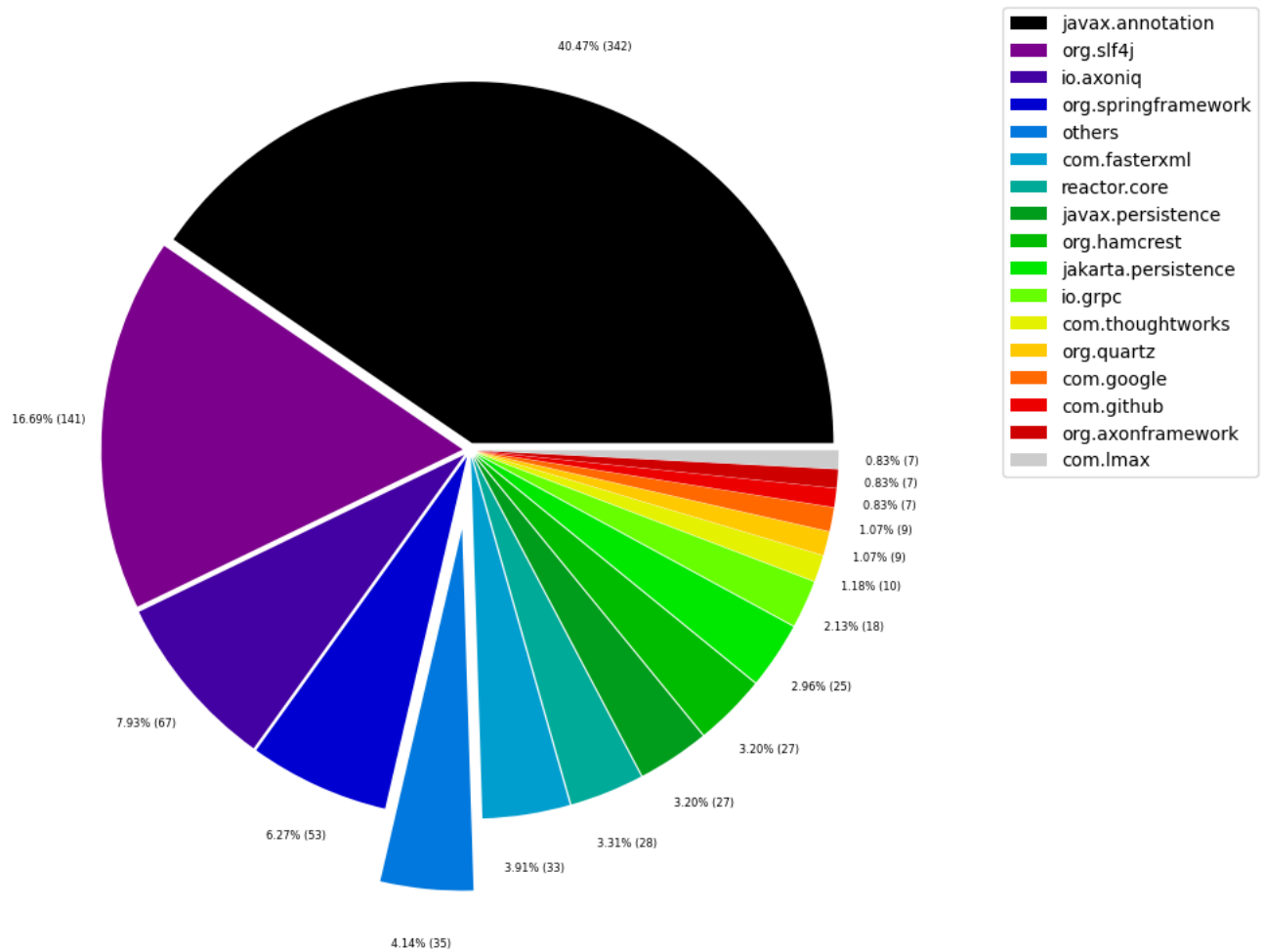


Table 2 Chart 1b - Most called second level external packages in % by type (less than 0.7% overall "others" drill-down)

Shows the lowest (less than 0.7% overall) most called external package. Therefore, this plot breaks down the "others" slice of the pie chart above. Values under 0.3% from that will be grouped into "others" to get a cleaner plot.

<Figure size 640x480 with 0 Axes>



Top external package (grouped by first 2 layers) usage [%] by type (less than 0.7% overall "others" drill-down)



Table 2 Chart 2a - Most called second level external packages in % by package (more than 0.7% overall)

External package groups that are used less than 0.7% are grouped into the name "others" to get a cleaner chart with the most significant external packages and how often they are called in percent.

<Figure size 640x480 with 0 Axes>

Top external package (grouped by first 2 layers) usage [%] by package (more than 0.7% overall)



Table 2 Chart 2b - Most called second level external packages in % by package (less than 0.7% overall "others" drill-down)

Shows the lowest (less than 0.7% overall) most called external package. Therefore, this plot breaks down the "others" slice of the pie chart above. Values under 0.3% from that will be grouped into "others" to get a cleaner plot.

<Figure size 640x480 with 0 Axes>

Top external package (grouped by first 2 layers) usage [%] by package (less than 0.7% overall "others" drill-down)

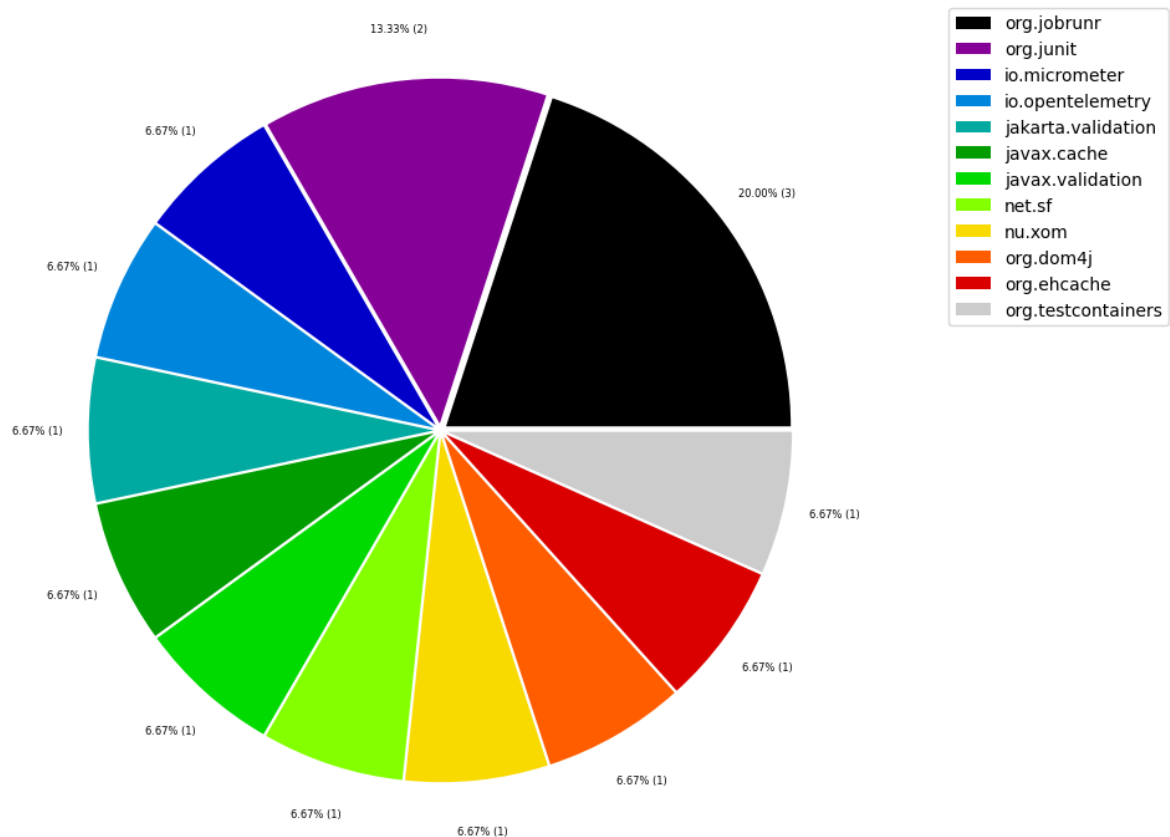


Table 3 - Top 20 most widely spread external packages

The following tables shows external packages that are used by many different artifacts with the highest number of artifacts first. External annotations are filtered out to only get those external packages that significantly add to coupling.

Statistics like minimum, maximum, average, median and standard deviation are provided for the number of packages and number of types in every artifact that uses the listed external package.

The intuition behind that is to find external package dependencies that are used in a widely spread manner. This should uncover libraries and frameworks and make it easier to distinguish them from external dependencies that are used for specific tasks. It can also be used to find external dependencies that are used sparsely regarding artifacts but are used in many different packages there. This could then be improved by applying a [Hexagonal architecture](#).

Only the top 20 entries are shown. The whole table can be found in the following CSV report:

External\_package\_usage\_spread

#### Columns:

- *externalPackageName* identifies the external package as defined above. All other columns contain aggregated data for this external package.
- *numberOfArtifacts* contains the number of artifacts that use the external package

- *sumNumberOfPackages* contains the sum of all packages that use the external package
- *min/max/med/avg/stdNumberOfPackages* provide statistics based on the number of packages of each artifact that uses the external package
- *min/max/med/avg/stdNumberOfPackagesPercentage* provide statistics in percent (%) based on the number of packages of each artifact that uses the external package
- *min/max/med/avg/stdNumberOfTypes* provide statistics based on the number of types of each artifact that uses the external package
- *min/max/med/avg/stdNumberOfPackagesPercentage* provide statistics in percent (%) based on the number of types of each artifact that uses the external package
- *someArtifactNames* contain some of the artifacts that contain the external package for reference

	externalPackageName	numberOfArtifacts	sumNumberOfPackages	minNumberOfPackages	maxNumberOfPackages	medNumberOfPackages	avgNum
0	org.slf4j	9	66	1	38	3.0	
1	jakarta.persistence	4	8	1	3	2.0	
2	javax.persistence	4	11	2	4	2.5	
3	com.fasterxml.jackson.databind	2	2	1	1	1.0	
4	com.github.kagkarlsson.scheduler	2	3	1	2	1.5	
5	com.github.kagkarlsson.scheduler.task	2	3	1	2	1.5	
6	com.thoughtworks.xstream	2	3	1	2	1.5	
7	io.axoniq.axonserver.connector.event	2	2	1	1	1.0	
8	io.axoniq.axonserver.connector.impl	2	4	1	3	2.0	
9	org.jobrunr.scheduling	2	3	1	2	1.5	

10 rows × 25 columns

Table 3a - Top 20 most widely spread external packages - number of internal packages

This table shows the top 20 most widely spread external packages focussing on the spread across the number of internal packages.

	externalPackageName	numberOfArtifacts	minNumberOfPackages	maxNumberOfPackages	medNumberOfPackages	avgNumberOfPackages	stdNu
0	org.slf4j	9	1	38	3.0	7.333333	
1	jakarta.persistence	4	1	3	2.0	2.000000	
2	javax.persistence	4	2	4	2.5	2.750000	
3	com.fasterxml.jackson.databind	2	1	1	1.0	1.000000	
4	com.github.kagkarlsson.scheduler	2	1	2	1.5	1.500000	
5	com.github.kagkarlsson.scheduler.task	2	1	2	1.5	1.500000	
6	com.thoughtworks.xstream	2	1	2	1.5	1.500000	
7	io.axoniq.axonserver.connector.event	2	1	1	1.0	1.000000	
8	io.axoniq.axonserver.connector.impl	2	1	3	2.0	2.000000	
9	org.jobrunr.scheduling	2	1	2	1.5	1.500000	
10	org.reactivestreams	2	1	2	1.5	1.500000	
11	reactor.core.publisher	2	2	2	2.0	2.000000	
12	AggregateEventPublisherImpl	1	1	1	1.0	1.000000	
13	WeakValue	1	1	1	1.0	1.000000	
14	com.codahale.metrics	1	1	1	1.0	1.000000	
15	com.fasterxml.jackson.annotation	1	3	3	3.0	3.000000	
16	com.fasterxml.jackson.core	1	1	1	1.0	1.000000	
17	com.fasterxml.jackson.databind.jsontype	1	1	1	1.0	1.000000	
18	com.fasterxml.jackson.databind.module	1	1	1	1.0	1.000000	
19	com.fasterxml.jackson.databind.node	1	1	1	1.0	1.000000	

Table 3b - Top 20 most widely spread external packages - percentage of internal packages

This table shows the top 20 most widely spread external packages focussing on the spread across the percentage of internal packages.

	externalPackageName	numberOfArtifacts	minNumberOfPackagesPercentage	maxNumberOfPackagesPercentage	medNumberOfPackagesPercentage
0	org.slf4j	9	25.000000	100.000000	60.000000
1	jakarta.persistence	4	4.687500	22.222222	15.555556
2	javax.persistence	4	6.250000	30.000000	22.222222
3	com.fasterxml.jackson.databind	2	1.562500	11.111111	6.336806
4	com.github.kagkarlsson.scheduler	2	3.125000	11.111111	7.118056
5	com.github.kagkarlsson.scheduler.task	2	3.125000	11.111111	7.118056
6	com.thoughtworks.xstream	2	3.125000	11.111111	7.118056
7	io.axoniq.axonserver.connector.event	2	9.090909	11.111111	10.101010
8	io.axoniq.axonserver.connector.impl	2	11.111111	27.272727	19.191919
9	org.jobrunr.scheduling	2	3.125000	11.111111	7.118056
10	org.reactivestreams	2	3.125000	9.090909	6.107955
11	reactor.core.publisher	2	3.125000	18.181818	10.653409
12	AggregateEventPublisherImpl	1	12.500000	12.500000	12.500000
13	WeakValue	1	100.000000	100.000000	100.000000
14	com.codahale.metrics	1	11.111111	11.111111	11.111111
15	com.fasterxml.jackson.annotation	1	4.687500	4.687500	4.687500
16	com.fasterxml.jackson.core	1	1.562500	1.562500	1.562500
17	com.fasterxml.jackson.databind.jsontype	1	1.562500	1.562500	1.562500
18	com.fasterxml.jackson.databind.module	1	1.562500	1.562500	1.562500
19	com.fasterxml.jackson.databind.node	1	1.562500	1.562500	1.562500

Table 3c - Top 20 most widely spread external packages - number of internal types

This table shows the top 20 most widely spread external packages focussing on the spread across the number of internal types.

	externalPackageName	numberOfArtifacts	minNumberOfTypes	maxNumberOfTypes	medNumberOfTypes	avgNumberOfTypes	stdNumberOfTypes
0	org.slf4j	9	1	78	8.0	15.666667	24.067613
1	jakarta.persistence	4	3	8	3.0	4.250000	2.500000
2	javax.persistence	4	3	8	3.0	4.250000	2.500000
3	com.fasterxml.jackson.databind	2	2	7	4.5	4.500000	3.535534
4	com.github.kagkarlsson.scheduler	2	1	4	2.5	2.500000	2.121320
5	com.github.kagkarlsson.scheduler.task	2	1	4	2.5	2.500000	2.121320
6	com.thoughtworks.xstream	2	2	4	3.0	3.000000	1.414214
7	io.axoniq.axonserver.connector.event	2	2	11	6.5	6.500000	6.363961
8	io.axoniq.axonserver.connector.impl	2	1	4	2.5	2.500000	2.121320
9	org.jobrunr.scheduling	2	1	4	2.5	2.500000	2.121320
10	org.reactivestreams	2	4	13	8.5	8.500000	6.363961
11	reactor.core.publisher	2	9	18	13.5	13.500000	6.363961
12	AggregateEventPublisherImpl	1	1	1	1.0	1.000000	0.000000
13	WeakValue	1	1	1	1.0	1.000000	0.000000
14	com.codahale.metrics	1	1	1	1.0	1.000000	0.000000
15	com.fasterxml.jackson.annotation	1	5	5	5.0	5.000000	0.000000
16	com.fasterxml.jackson.core	1	3	3	3.0	3.000000	0.000000
17	com.fasterxml.jackson.databind.jsontype	1	1	1	1.0	1.000000	0.000000
18	com.fasterxml.jackson.databind.module	1	1	1	1.0	1.000000	0.000000
19	com.fasterxml.jackson.databind.node	1	2	2	2.0	2.000000	0.000000

Table 3d - Top 20 most widely spread external packages - percentage of internal types

This table shows the top 20 most widely spread external packages focussing on the spread across the percentage of internal types.

	externalPackageName	numberOfArtifacts	minNumberOfTypesPercentage	maxNumberOfTypesPercentage	medNumberOfTypesPercentage	avgNumt
0	org.slf4j	9	2.298851	36.363636	10.526316	
1	jakarta.persistence	4	1.016518	4.054054	2.089358	
2	javax.persistence	4	1.016518	4.054054	2.089358	
3	com.fasterxml.jackson.databind	2	0.889454	2.702703	1.796078	
4	com.github.kagkarlsson.scheduler	2	0.508259	1.351351	0.929805	
5	com.github.kagkarlsson.scheduler.task	2	0.508259	1.351351	0.929805	
6	com.thoughtworks.xstream	2	0.508259	2.702703	1.605481	
7	io.axoniq.axonserver.connector.event	2	2.702703	8.396947	5.549825	
8	io.axoniq.axonserver.connector.impl	2	1.351351	3.053435	2.202393	
9	org.jobrunr.scheduling	2	0.508259	1.351351	0.929805	
10	org.reactivestreams	2	1.651842	3.053435	2.352639	
11	reactor.core.publisher	2	2.287166	6.870229	4.578698	
12	AggregateEventPublisherImpl	1	1.149425	1.149425	1.149425	
13	WeakValue	1	4.545455	4.545455	4.545455	
14	com.codahale.metrics	1	1.351351	1.351351	1.351351	
15	com.fasterxml.jackson.annotation	1	0.635324	0.635324	0.635324	
16	com.fasterxml.jackson.core	1	0.381194	0.381194	0.381194	
17	com.fasterxml.jackson.databind.jsontype	1	0.127065	0.127065	0.127065	
18	com.fasterxml.jackson.databind.module	1	0.127065	0.127065	0.127065	
19	com.fasterxml.jackson.databind.node	1	0.254130	0.254130	0.254130	

Table 3 Chart 1a - Most widely spread external packages in % by types (more than 0.5% overall)

External packages that are used less than 0.5% are grouped into the name "others" to get a cleaner chart with the most significant external packages.

<Figure size 640x480 with 0 Axes>

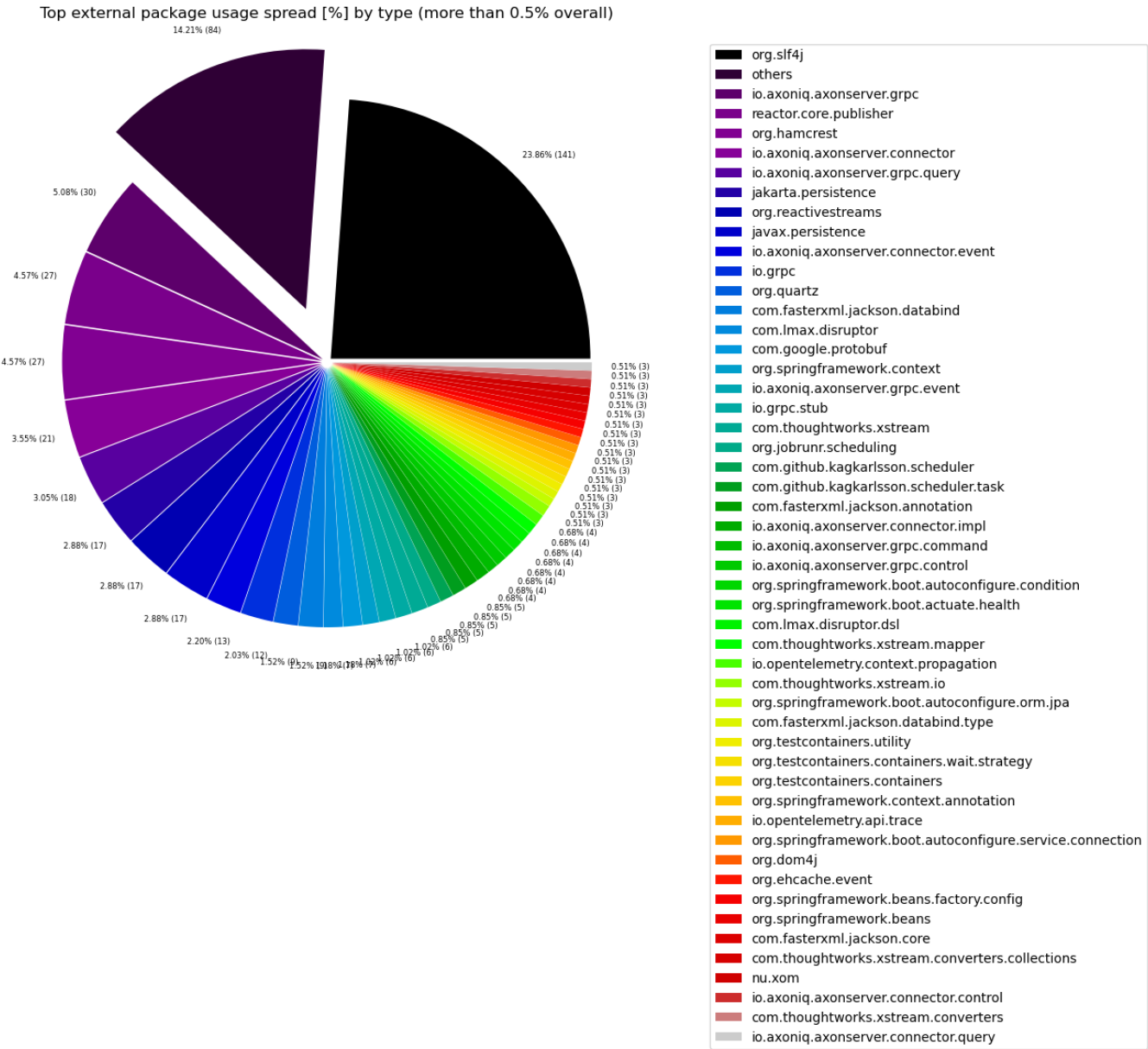


Table 3 Chart 1b - Most widely spread external packages in % by types (less than 0.5% overall "others" drill-down)

Shows the lowest (less than 0.5% overall) most spread external packages. Therefore, this plot breaks down the "others" slice of the pie chart above. Values under 0.3% from that will be grouped into "others" to get a cleaner plot.

<Figure size 640x480 with 0 Axes>

Top external package usage spread [%] by type (less than 0.5% overall "others" drill-down)

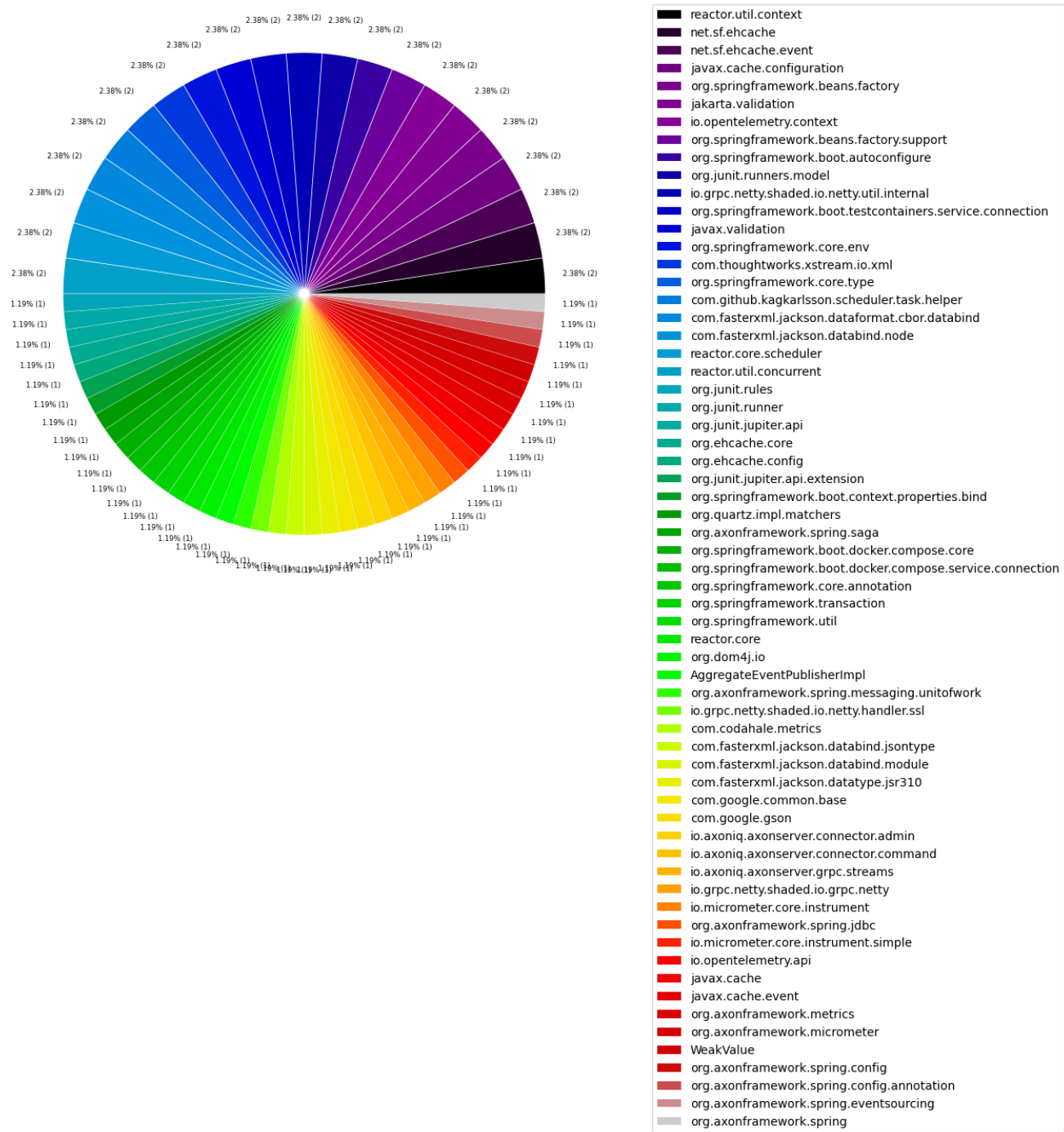


Table 3 Chart 2a - Most widely spread external packages in % by packages (more than 0.5% overall)

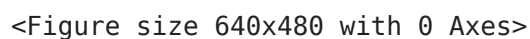
External packages that are used less than 0.5% are grouped into the name "others" to get a cleaner chart with the most significant external packages.

<Figure size 640x480 with 0 Axes>



A pie chart illustrating the distribution of 100 categories. The largest category is black, representing 30.20% (77). The second largest is dark purple, representing 25.88% (66). Other categories include 4.31% (11) in purple, 3.14% (8) in magenta, 2.75% (7) in blue-purple, 2.35% (6) in blue, 1.96% (5) in teal, 1.57% (4) in green, 1.18% (3) in yellow-green, and 0.78% (2) in yellow and orange. The remaining categories are very small, each representing 0.78% (2) or less.

Category Color	Percentage	Count
Black	30.20%	77
Dark Purple	25.88%	66
Purple	4.31%	11
Magenta	3.14%	8
Blue-Purple	2.75%	7
Blue	2.35%	6
Teal	1.96%	5
Green	1.57%	4
Yellow-Green	1.18%	3
Yellow	0.78%	2
Orange	0.78%	2
Red	0.78%	2
Pink	0.78%	2
Light Purple	0.78%	2
Light Blue	0.78%	2
Light Green	0.78%	2
Light Yellow	0.78%	2
White	0.78%	2



Top external package usage spread [%] by type (less than 0.7% overall "others" drill-down)

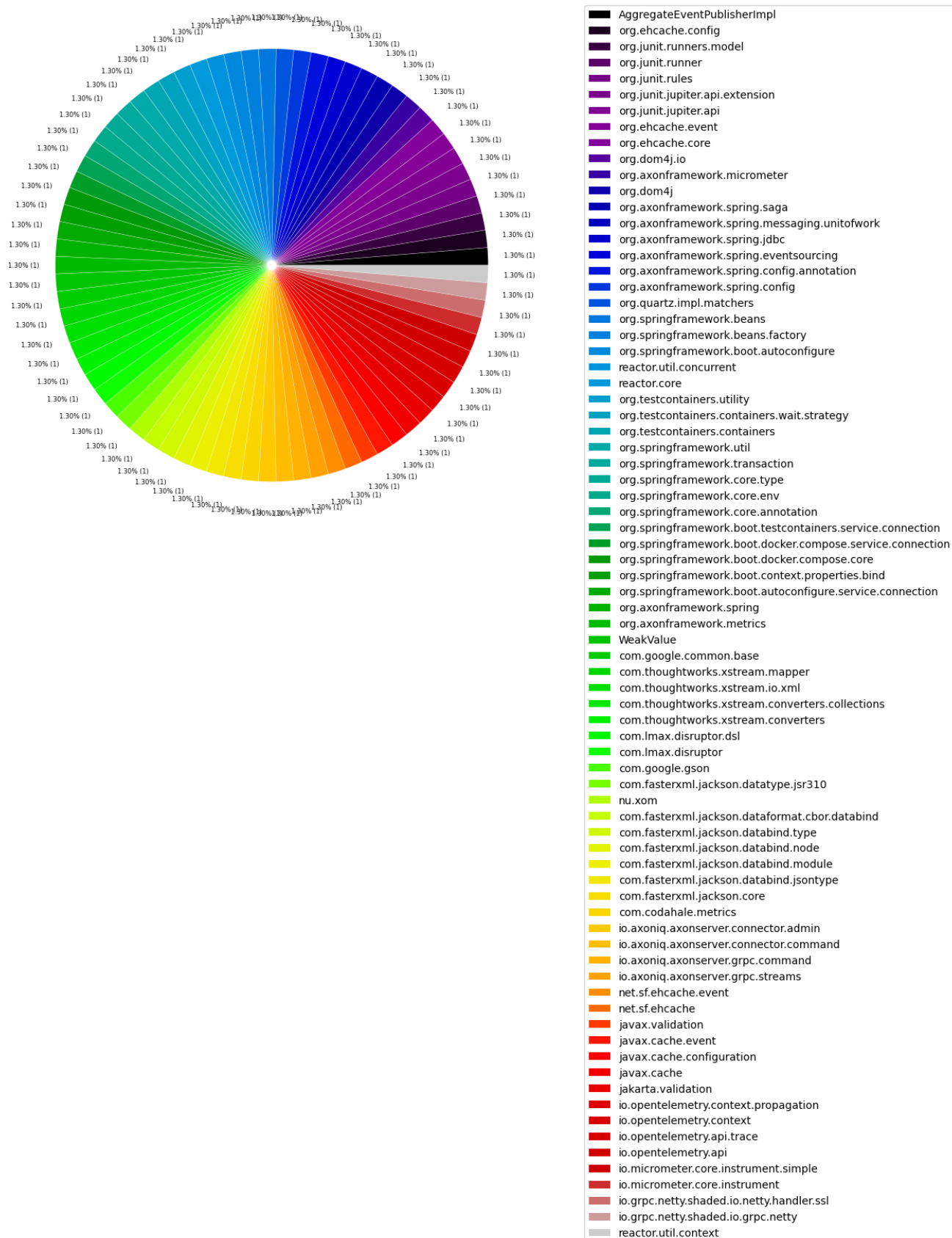


Table 4 - Top 20 most widely spread external packages grouped by their first 2 layers

This table shows external packages grouped by their first 2 layers that are used by many different artifacts with the highest number of artifacts first. External annotations are filtered out to only get those external packages that significantly add to coupling.

Statistics like minimum, maximum, average, median and standard deviation are provided for the number of packages and number of types in every artifact that uses the listed external package.

The intuition behind that is to find external package dependencies that are used in a widely spread manner. This should uncover libraries and frameworks and make it easier to distinguish them from external dependencies that are used for specific tasks. It can also be used to find external dependencies that are used sparsely regarding artifacts but are used in many different packages there. This could then be improved by applying a [Hexagonal architecture](#).

Only the top 20 entries are shown. The whole table can be found in the following CSV report:

`External_package_usage_spread`

#### Columns:

- *externalPackageName* identifies the external package as defined above. All other columns contain aggregated data for this external package.
- *numberOfArtifacts* contains the number of artifacts that use the external package
- *sumNumberOfPackages* contains the sum of all packages that use the external package
- *min/max/med/avg/stdNumberOfPackages* provide statistics based on the number of packages of each artifact that uses the external package
- *min/max/med/avg/stdNumberOfPackagesPercentage* provide statistics in percent (%) based on the number of packages of each artifact that uses the external package
- *min/max/med/avg/stdNumberOfTypes* provide statistics based on the number of types of each artifact that uses the external package
- *min/max/med/avg/stdNumberOfPackagesPercentage* provide statistics in percent (%) based on the number of types of each artifact that uses the external package
- *someArtifactNames* contain some of the artifacts that contain the external package for reference

	externalSecondLevelPackageName	numberOfArtifacts	sumNumberOfPackages	minNumberOfPackages	maxNumberOfPackages	medNumberOfPackages	avgNum
0	org.slf4j	9	66	1	38	3.0	
1	jakarta.persistence	4	8	1	3	2.0	
2	javax.persistence	4	11	2	4	2.5	
3	com.fasterxml	2	5	1	4	2.5	
4	com.github	2	3	1	2	1.5	
5	com.google	2	4	1	3	2.0	
6	com.thoughtworks	2	3	1	2	1.5	
7	io.axoniq	2	9	1	8	4.5	
8	org.jobrunr	2	3	1	2	1.5	
9	org.reactivestreams	2	3	1	2	1.5	
10	reactor.core	2	5	2	3	2.5	
11	AggregateEventPublisherImpl	1	1	1	1	1.0	
12	WeakValue	1	1	1	1	1.0	
13	com.codahale	1	1	1	1	1.0	
14	com.lmax	1	1	1	1	1.0	
15	io.grpc	1	5	5	5	5.0	
16	io.micrometer	1	1	1	1	1.0	
17	io.opentelemetry	1	1	1	1	1.0	
18	jakarta.validation	1	1	1	1	1.0	
19	javax.cache	1	1	1	1	1.0	

20 rows × 25 columns

Table 4 Chart 1a - Most widely spread second level external packages in % by type (more than 0.5% overall)

External package groups that are used less than 0.5% are grouped into the name "others" to get a cleaner chart with the most significant external packages and how often they are called in percent.

<Figure size 640x480 with 0 Axes>

Top external package (grouped by first 2 layers) usage spread [%] by type

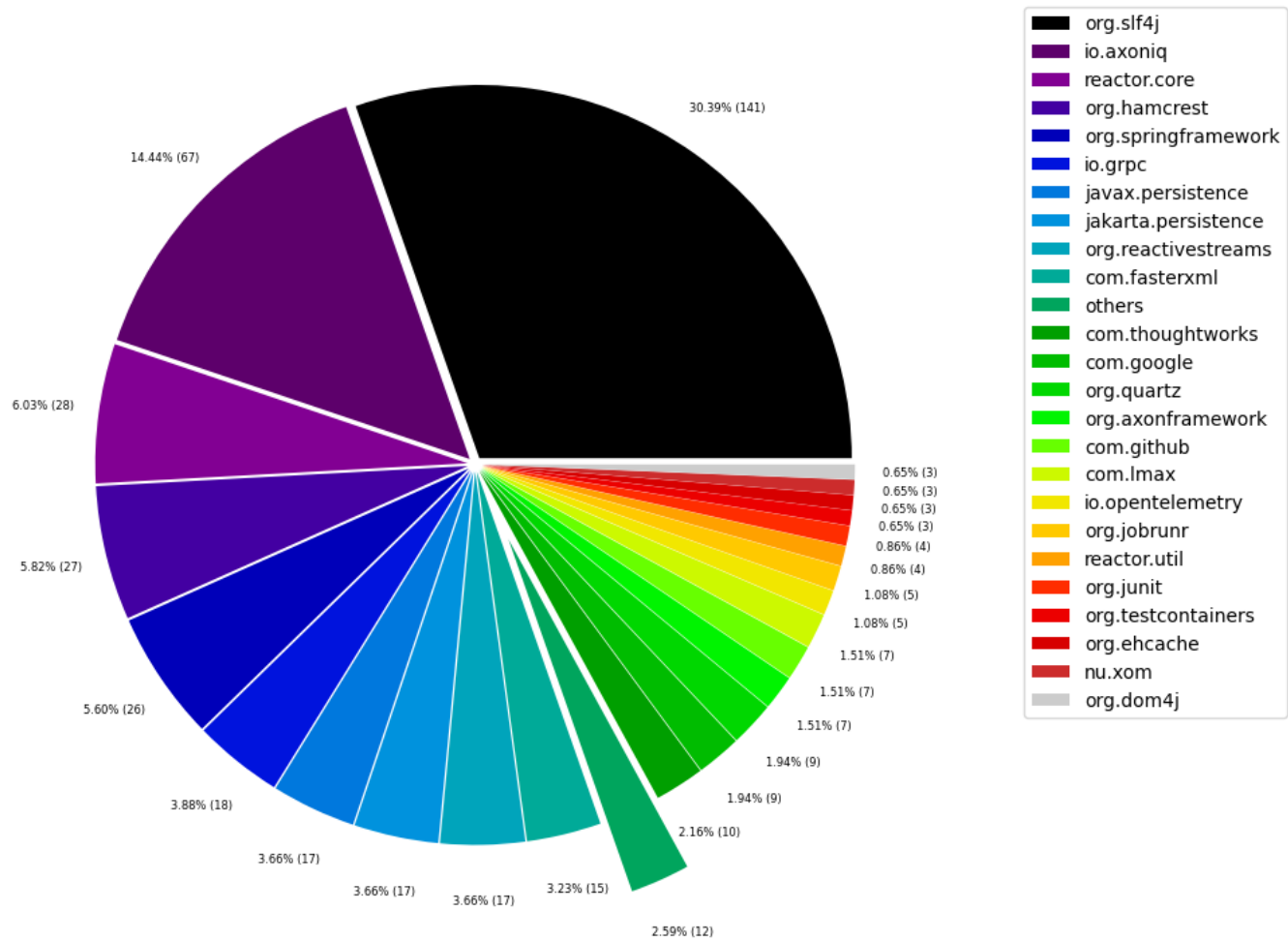


Table 4 Chart 1b - Most widely spread second level external packages in % by type (less than 0.5% overall "others" drill-down)

External packages that are used less than 0.5% are grouped into the name "others" to get a cleaner chart with the most significant external packages.

<Figure size 640x480 with 0 Axes>

Top external package usage spread [%] by type (less than 0.7% overall "others" drill-down)



Table 4 Chart 2a - Most widely spread second level external packages in % by package (more than 0.5% overall)

External package groups that are used less than 0.5% are grouped into the name "others" to get a cleaner chart with the most significant external packages and how often they are called in percent.

<Figure size 640x480 with 0 Axes>

Top external package (grouped by first 2 layers) usage spread [%] by package (more than 0.5% overall)



Table 4 Chart 2b - Most widely spread second level external packages in % by package (less than 0.5% overall "others" drill-down)

External packages that are used less than 0.5% are grouped into the name "others" to get a cleaner chart with the most significant external packages.

No data to plot for title 'Top external package (less than 0.7% overall "others" drill-down) '.

Table 5 - Top 20 least used external packages overall

This table identifies external packages that aren't used very often. This could help to find libraries that aren't actually needed or maybe easily replaceable. Some of them might be used sparsely on purpose for example as an adapter to an external library that is actually important. Thus, decisions need to be made on a case-by-case basis.

Only the last 20 entries are shown. The whole table can be found in the following CSV report:

External\_package\_usage\_overall

Columns:

- externalPackageName identifies the external package as described above
- numberOfExternalTypeCalls includes every invocation or reference to the types in the external package

	externalPackageName	numberOfExternalTypeCalls
0	org.springframework.boot.docker.compose.servic...	2
1	org.springframework.boot.docker.compose.core	2
2	org.axonframework.spring.eventsourcing	2
3	org.axonframework.spring.config.annotation	2
4	org.axonframework.micrometer	2
5	org.axonframework.metrics	2
6	reactor.core.scheduler	3
7	org.springframework.boot.context.properties.bind	3
8	com.fasterxml.jackson.databind.node	3
9	org.junit.jupiter.api.extension	3
10	net.sf.ehcache.event	3
11	javax.cache.configuration	3
12	com.fasterxml.jackson.core	4
13	org.testcontainers.containers.wait.strategy	4
14	com.google.gson	4
15	com.thoughtworks.xstream.io.xml	4
16	org.springframework.beans.factory.config	4
17	org.springframework.beans.factory.support	4
18	org.springframework.beans.factory	4
19	org.springframework.boot.testcontainers.servic...	4

**Table 6 - External usage per artifact sorted by highest external type rate descending**

The following table shows the most used external packages separately for each artifact including external annotations. The results are sorted by the artifacts with the highest external type usage rate descending.

The intention of this table is to find artifacts that use a lot of external dependencies in relation to their size and get all the external packages and their usage.

Only the last 40 entries are shown. The whole table can be found in the following CSV report:

[External\\_package\\_usage\\_per\\_artifact\\_sorted](#)

#### Columns:

- *artifactName* is used to group the the external package usage per artifact for a more detailed analysis.
- *externalPackageName* identifies the external package as described above
- *numberOfExternalTypeCaller* refers to the distinct types that make use of the external package
- *numberOfExternalTypeCalls* includes every invocation or reference to the types in the external package
- *numberOfTypesInArtifact* represents the total count of all analyzed types for the artifact
- *numberOfExternalTypesInArtifact* is the number of all external types that are used by the artifact
- *numberOfExternalPackagesInArtifact* is the number of all external packages that are used by the artifact
- *externalTypeRate* is the  $\text{numberOfExternalTypesInArtifact} / \text{numberOfTypesInArtifact} * 100$
- *externalTypeNames* contains a list of actually utilized types of the external package





	artifactName	externalPackageName	numberOfExternalTypeCaller	numberOfExternalTypeCalls	numberOfTypesInArtifact	numberOfExtern
0	axon-tracing-opentelemetry-4.10.0	io.opentelemetry.api.trace	9	47	5	
1	axon-tracing-opentelemetry-4.10.0	io.opentelemetry.context.propagation	9	18	5	
2	axon-tracing-opentelemetry-4.10.0	javax.annotation	3	8	5	
3	axon-tracing-opentelemetry-4.10.0	io.opentelemetry.context	2	7	5	
4	axon-tracing-opentelemetry-4.10.0	org.slf4j	2	7	5	
5	axon-tracing-opentelemetry-4.10.0	io.opentelemetry.api	1	2	5	
6	axon-spring-boot-autoconfigure-4.10.0	org.springframework.boot.autoconfigure.condition	66	150	74	
7	axon-spring-boot-autoconfigure-4.10.0	org.springframework.boot.autoconfigure	54	55	74	
8	axon-spring-boot-autoconfigure-4.10.0	org.springframework.context.annotation	41	134	74	
9	axon-spring-boot-autoconfigure-4.10.0	org.springframework.boot.context.properties	18	18	74	
10	axon-spring-boot-autoconfigure-4.10.0	org.springframework.beans.factory.annotation	11	28	74	
11	axon-spring-boot-autoconfigure-4.10.0	javax.annotation	8	8	74	
12	axon-spring-boot-autoconfigure-4.10.0	org.slf4j	8	13	74	
13	axon-spring-boot-autoconfigure-4.10.0	org.springframework.context	8	29	74	
14	axon-spring-boot-autoconfigure-4.10.0	org.springframework.boot.actuate.health	7	24	74	
15	axon-spring-boot-autoconfigure-4.10.0	org.axonframework.spring.config	5	17	74	
16	axon-spring-boot-autoconfigure-4.10.0	jakarta.persistence	4	6	74	
17	axon-spring-boot-autoconfigure-4.10.0	javax.persistence	4	6	74	
18	axon-spring-boot-autoconfigure-4.10.0	org.springframework.beans.factory	4	14	74	
19	axon-spring-boot-autoconfigure-4.10.0	org.springframework.beans.factory.config	4	7	74	
20	axon-spring-boot-autoconfigure-4.10.0	org.springframework.beans.factory.support	4	21	74	
21	axon-spring-boot-autoconfigure-4.10.0	org.springframework.boot.testcontainers.servic...	4	9	74	

	artifactName	externalPackageName	numberOfExternalTypeCaller	numberOfExternalTypeCalls	numberOfTypesInArtifact	numberOfExtern
22	axon-spring-boot-autoconfigure-4.10.0	org.springframework.core.annotation	4	4	74	
23	axon-spring-boot-autoconfigure-4.10.0	org.springframework.beans	3	4	74	
24	axon-spring-boot-autoconfigure-4.10.0	org.springframework.boot.autoconfigure.orm.jpa	3	3	74	
25	axon-spring-boot-autoconfigure-4.10.0	org.springframework.boot.autoconfigure.service...	3	3	74	
26	axon-spring-boot-autoconfigure-4.10.0	org.springframework.boot.context.properties.bind	3	4	74	
27	axon-spring-boot-autoconfigure-4.10.0	com.fasterxml.jackson.databind	2	10	74	
28	axon-spring-boot-autoconfigure-4.10.0	com.fasterxml.jackson.dataformat.cbor.databind	2	8	74	
29	axon-spring-boot-autoconfigure-4.10.0	com.thoughtworks.xstream	2	8	74	
30	axon-spring-boot-autoconfigure-4.10.0	io.axoniq.axonserver.connector.event	2	2	74	
31	axon-spring-boot-autoconfigure-4.10.0	org.axonframework.metrics	2	8	74	
32	axon-spring-boot-autoconfigure-4.10.0	org.axonframework.micrometer	2	10	74	
33	axon-spring-boot-autoconfigure-4.10.0	org.axonframework.spring.config.annotation	2	7	74	
34	axon-spring-boot-autoconfigure-4.10.0	org.axonframework.spring.eventsourcing	2	10	74	
35	axon-spring-boot-autoconfigure-4.10.0	org.springframework.boot.docker.compose.core	2	5	74	
36	axon-spring-boot-autoconfigure-4.10.0	org.springframework.boot.docker.compose.servic...	2	5	74	
37	axon-spring-boot-autoconfigure-4.10.0	org.springframework.core.env	2	2	74	
38	axon-spring-boot-autoconfigure-4.10.0	org.springframework.core.type	2	4	74	
39	axon-spring-boot-autoconfigure-4.10.0	com.codahale.metrics	1	6	74	

**Table 7 - Artifacts and their external packages**

The following table shows the artifacts with the highest external dependency usage broken down by each external package including external annotations. The results are sorted by the artifacts with the highest external package usage rate descending.

The intention of this table is to find artifacts that use a lot of external dependencies and show in detail which external packages are used by them and how many internal packages.

Only the last 30 entries are shown. The whole table can be found in the following CSV report:

`External_package_usage_per_artifact_and_external_package`

### Columns:

- *artifactName* is the name of the artifact with external dependencies (first grouping column)
- *artifactPackages* is the number of packages in the artifact
- *artifactTypes* is the number of types in the artifact
- *artifactExternalPackages* is the number of external packages used by the artifact
- *artifactExternalCallingPackages* is the number of packages that use external packages in the artifact
- *artifactExternalCallingPackagesRate* is  $\text{artifactExternalCallingPackages} / \text{artifactPackages} * 100\%$
- *externalPackageName* the name of the external package (second grouping column)
- *numberOfPackages* is the number of internal packages of the artifact that use the external packages
- *numberOfTypes* is the number of internal types of the artifact that use the external packages
- *packagesCallingExternalRate* is  $\text{numberOfPackages} / \text{artifactPackages} * 100\%$
- *typesCallingExternalRate* is  $\text{numberOfTypes} / \text{artifactTypes} * 100\%$
- *nameOfPackages* names of the internal packages that use the external package in the artifact
- *someTypeNames* some (10) names of the internal types that use the external package in the artifact

	artifactName	artifactPackages	artifactTypes	artifactExternalPackages	artifactExternalCallingPackages	artifactExternalCallingPackagesRate	
0	axon-configuration-4.10.0	1	41	2	1	100.0	
1	axon-configuration-4.10.0	1	41	2	1	100.0	
2	axon-disruptor-4.10.0	1	22	5	1	100.0	
3	axon-disruptor-4.10.0	1	22	5	1	100.0	
4	axon-disruptor-4.10.0	1	22	5	1	100.0	
5	axon-disruptor-4.10.0	1	22	5	1	100.0	
6	axon-disruptor-4.10.0	1	22	5	1	100.0	
7	axon-spring-boot-autoconfigure-4.10.0	9	74	47	9	100.0	
8	axon-spring-boot-autoconfigure-4.10.0	9	74	47	9	100.0	
9	axon-spring-boot-autoconfigure-4.10.0	9	74	47	9	100.0	org.springframework
10	axon-spring-boot-autoconfigure-4.10.0	9	74	47	9	100.0	org.springframework
11	axon-spring-boot-autoconfigure-4.10.0	9	74	47	9	100.0	org.springframework.boot
12	axon-spring-boot-autoconfigure-4.10.0	9	74	47	9	100.0	org.springframework
13	axon-spring-boot-autoconfigure-4.10.0	9	74	47	9	100.0	org.springframework
14	axon-spring-boot-autoconfigure-4.10.0	9	74	47	9	100.0	
15	axon-spring-boot-autoconfigure-4.10.0	9	74	47	9	100.0	
16	axon-spring-boot-autoconfigure-4.10.0	9	74	47	9	100.0	org.springframework
17	axon-spring-boot-autoconfigure-4.10.0	9	74	47	9	100.0	org.springframework
18	axon-spring-boot-autoconfigure-4.10.0	9	74	47	9	100.0	org.springframework
19	axon-spring-boot-autoconfigure-4.10.0	9	74	47	9	100.0	org.springframework
20	axon-spring-boot-autoconfigure-4.10.0	9	74	47	9	100.0	org.springframework
21	axon-spring-boot-autoconfigure-4.10.0	9	74	47	9	100.0	org.springframework

	artifactName	artifactPackages	artifactTypes	artifactExternalPackages	artifactExternalCallingPackages	artifactExternalCallingPackagesRate	
22	axon-spring-boot-autoconfigure-4.10.0	9	74	47	9	100.0	
23	axon-spring-boot-autoconfigure-4.10.0	9	74	47	9	100.0	com.
24	axon-spring-boot-autoconfigure-4.10.0	9	74	47	9	100.0	com.fasterxml.jackso
25	axon-spring-boot-autoconfigure-4.10.0	9	74	47	9	100.0	com.git
26	axon-spring-boot-autoconfigure-4.10.0	9	74	47	9	100.0	com.github.l
27	axon-spring-boot-autoconfigure-4.10.0	9	74	47	9	100.0	,
28	axon-spring-boot-autoconfigure-4.10.0	9	74	47	9	100.0	io.axoniq.ε
29	axon-spring-boot-autoconfigure-4.10.0	9	74	47	9	100.0	io.axoniq

Table 7a - Artifacts and their external packages (first 2 levels)

The following table groups the external packages by their first two levels. For example

`javax.xml.namespace` and `javax.xml.stream` will be grouped together to `javax.xml`.

	artifactName	artifactPackages	artifactTypes	artifactExternalPackagesFirst2Levels	artifactExternalCallingPackages	artifactExternalCallingPackagesRate	externalI
0	axon-configuration-4.10.0	1	41	2	1	100.00	
1	axon-configuration-4.10.0	1	41	2	1	100.00	
2	axon-disruptor-4.10.0	1	22	4	1	100.00	
3	axon-disruptor-4.10.0	1	22	4	1	100.00	
4	axon-disruptor-4.10.0	1	22	4	1	100.00	
5	axon-disruptor-4.10.0	1	22	4	1	100.00	
6	axon-spring-boot-autoconfigure-4.10.0	9	74	13	9	100.00	
7	axon-spring-boot-autoconfigure-4.10.0	9	74	13	9	100.00	
8	axon-spring-boot-autoconfigure-4.10.0	9	74	13	9	100.00	
9	axon-spring-boot-autoconfigure-4.10.0	9	74	13	9	100.00	
10	axon-spring-boot-autoconfigure-4.10.0	9	74	13	9	100.00	
11	axon-spring-boot-autoconfigure-4.10.0	9	74	13	9	100.00	
12	axon-spring-boot-autoconfigure-4.10.0	9	74	13	9	100.00	
13	axon-spring-boot-autoconfigure-4.10.0	9	74	13	9	100.00	
14	axon-spring-boot-autoconfigure-4.10.0	9	74	13	9	100.00	
15	axon-spring-boot-autoconfigure-4.10.0	9	74	13	9	100.00	
16	axon-spring-boot-autoconfigure-4.10.0	9	74	13	9	100.00	
17	axon-spring-boot-autoconfigure-4.10.0	9	74	13	9	100.00	
18	axon-spring-boot-autoconfigure-4.10.0	9	74	13	9	100.00	
19	axon-tracing-opentelemetry-4.10.0	1	5	3	1	100.00	
20	axon-tracing-opentelemetry-4.10.0	1	5	3	1	100.00	
21	axon-tracing-opentelemetry-4.10.0	1	5	3	1	100.00	
22	axon-server-connector-4.10.0	11	131	8	10	90.91	
23	axon-server-connector-4.10.0	11	131	8	10	90.91	

	artifactName	artifactPackages	artifactTypes	artifactExternalPackagesFirst2Levels	artifactExternalCallingPackages	artifactExternalCallingPackagesRate	external
24	axon-server-connector-4.10.0	11	131	8	10	90.91	
25	axon-server-connector-4.10.0	11	131	8	10	90.91	
26	axon-server-connector-4.10.0	11	131	8	10	90.91	
27	axon-server-connector-4.10.0	11	131	8	10	90.91	
28	axon-server-connector-4.10.0	11	131	8	10	90.91	
29	axon-server-connector-4.10.0	11	131	8	10	90.91	

Table 7b - Top 15 external dependency using artifacts as columns with their external packages

The following table uses pivot to show the artifacts in columns, the external dependencies in rows and the number of internal packages as values.

artifactName	axon-messaging-4.10.0	axon-spring-boot-autoconfigure-4.10.0	axon-server-connector-4.10.0	axon-test-4.10.0	axon-modelling-4.10.0	axon-eventsourcing-4.10.0	axon-tracing-opentelemetry-4.10.0	axon-disruptor-4.10.0	axon-configuration-4.10.0
externalPackageName									
AggregateEventPublisherImpl	0	0	0	1	0	0	0	0	0
WeakValue	0	0	0	0	0	0	0	1	0
com.codahale.metrics	0	1	0	0	0	0	0	0	0
com.fasterxml.jackson.annotation	11	0	0	0	2	0	0	0	0
com.fasterxml.jackson.core	1	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...
reactor.core	1	0	0	0	0	0	0	0	0
reactor.core.publisher	2	0	2	0	0	0	0	0	0
reactor.core.scheduler	0	0	2	0	0	0	0	0	0
reactor.util.concurrent	1	0	0	0	0	0	0	0	0
reactor.util.context	1	0	0	0	0	0	0	0	0

117 rows × 9 columns

Table 7c - Top 15 external dependency using artifacts as columns with their external packages (first 2 levels)

The following table uses pivot to show the artifacts in columns, the external package name grouped by its first two levels in rows and the number of internal packages as values. For example

`javax.xml.namespace` and `javax.xml.stream` will be grouped together to `javax.xml`.



artifactName	axon-messaging-4.10.0	axon-server-connector-4.10.0	axon-spring-boot-autoconfigure-4.10.0	axon-modelling-4.10.0	axonsourcing-4.10.0	axon-test-4.10.0	axon-disruptor-4.10.0	axon-tracing-opentelemetry-4.10.0	axon-configuration-4.10.0
externalPackageNameFirst2Levels									
AggregateEventPublisherImpl	0	0	0	0	0	1	0	0	0
WeakValue	0	0	0	0	0	0	1	0	0
com.codahale	0	0	1	0	0	0	0	0	0
com.fasterxml	12	0	1	2	0	0	0	0	0
com.github	2	0	1	0	0	0	0	0	0
com.google	0	3	0	0	0	1	0	0	0
com.lmax	0	0	0	0	0	0	1	0	0
com.thoughtworks	2	0	1	0	0	0	0	0	0
io.axoniq	0	8	1	0	0	0	0	0	0
io.grpc	0	5	0	0	0	0	0	0	0
io.micrometer	0	0	1	0	0	0	0	0	0
io.opentelemetry	0	0	0	0	0	0	0	1	0
jakarta.persistence	4	0	2	2	2	0	0	0	0
jakarta.validation	1	0	0	0	0	0	0	0	0
javax.annotation	47	8	4	4	6	4	1	1	1
javax.cache	1	0	0	0	0	0	0	0	0
javax.persistence	7	0	2	3	3	0	0	0	0
javax.validation	1	0	0	0	0	0	0	0	0
net.sf	1	0	0	0	0	0	0	0	0
nu.xom	1	0	0	0	0	0	0	0	0
org.axonframework	0	0	2	0	0	0	0	0	0
org.dom4j	1	0	0	0	0	0	0	0	0
org.ehcache	1	0	0	0	0	0	0	0	0
org.hamcrest	0	0	0	0	0	5	0	0	0
org.jobrunr	2	0	1	0	0	0	0	0	0
org.junit	0	0	0	0	0	2	0	0	0
org.quartz	2	0	0	0	0	0	0	0	0
org.reactivestreams	2	1	0	0	0	0	0	0	0
org.slf4j	38	9	3	6	5	2	1	1	1
org.springframework	0	1	7	0	0	0	0	0	0
org.testcontainers	0	0	0	0	0	1	0	0	0
reactor.core	2	3	0	0	0	0	0	0	0
reactor.util	1	0	0	0	0	0	0	0	0

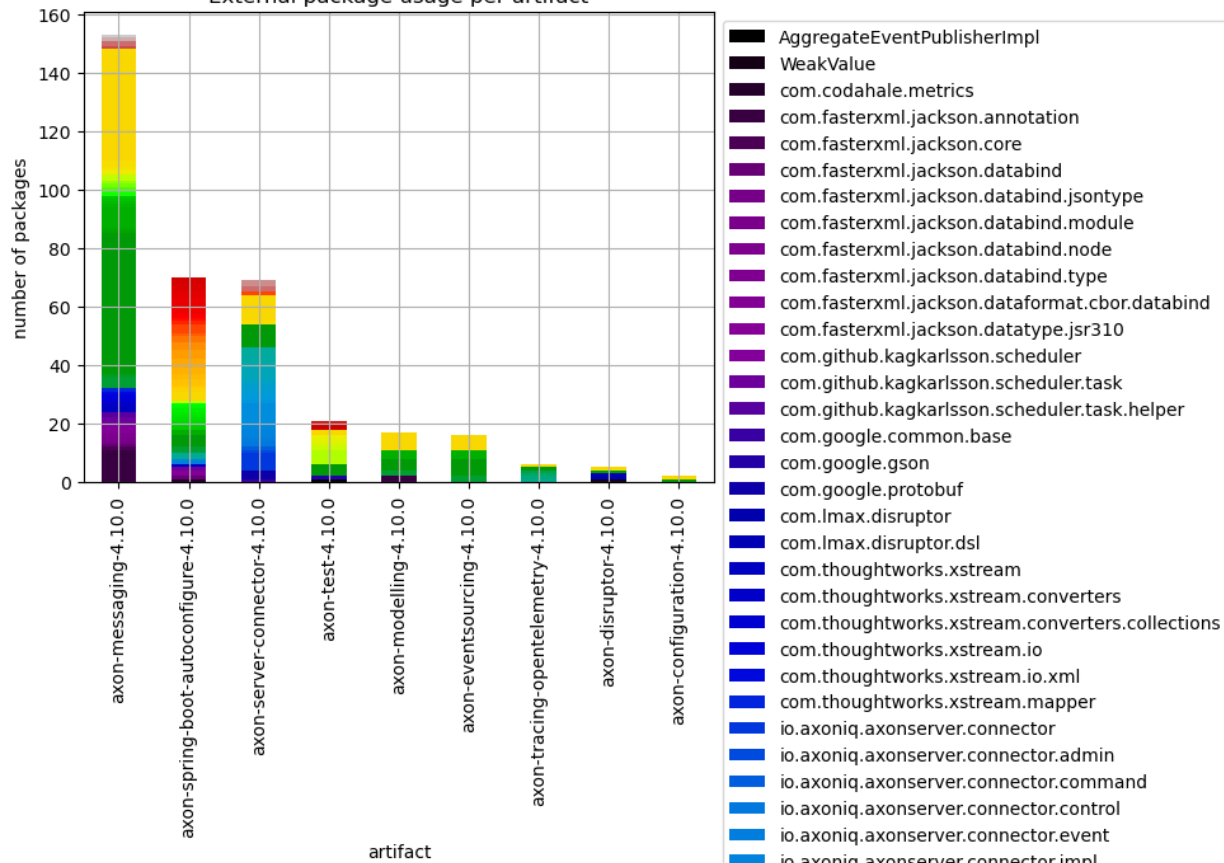
Table 7 Chart 1 - Top 15 external dependency using artifacts and their external packages stacked

The following chart shows the top 15 external package using artifacts and breaks down which external packages they use in how many different internal packages with stacked bars.

Note that every external dependency is counted separately so that if on internal package uses two external packages it will be displayed for both and so stacked twice.

<Figure size 640x480 with 0 Axes>

External package usage per artifact



- AggregateEventPublisherImpl
- WeakValue
- com.codahale.metrics
- com.fasterxml.jackson.annotation
- com.fasterxml.jackson.core
- com.fasterxml.jackson.databind
- com.fasterxml.jackson.databind.jsontype
- com.fasterxml.jackson.databind.module
- com.fasterxml.jackson.databind.node
- com.fasterxml.jackson.databind.type
- com.fasterxml.jackson.dataformat.cbor.databind
- com.fasterxml.jackson.datatype.jsr310
- com.github.kagkarlsson.scheduler
- com.github.kagkarlsson.scheduler.task
- com.github.kagkarlsson.scheduler.task.helper
- com.google.common.base
- com.google.gson
- com.google.protobuf
- com.lmax.disruptor
- com.lmax.disruptor.dsl
- com.thoughtworks.xstream
- com.thoughtworks.xstream.converters
- com.thoughtworks.xstream.converters.collections
- com.thoughtworks.xstream.io
- com.thoughtworks.xstream.io.xml
- com.thoughtworks.xstream.mapper
- io.axoniq.axonserver.connector
- io.axoniq.axonserver.connector.admin
- io.axoniq.axonserver.connector.command
- io.axoniq.axonserver.connector.control
- io.axoniq.axonserver.connector.event
- io.axoniq.axonserver.connector.impl
- io.axoniq.axonserver.connector.query
- io.axoniq.axonserver.grpc
- io.axoniq.axonserver.grpc.command
- io.axoniq.axonserver.grpc.control
- io.axoniq.axonserver.grpc.event
- io.axoniq.axonserver.grpc.query
- io.axoniq.axonserver.grpc.streams
- io.grpc
- io.grpc.netty.shaded.io.grpc.netty
- io.grpc.netty.shaded.io.netty.handler.ssl
- io.grpc.netty.shaded.io.netty.util.internal
- io.grpc.stub
- io.micrometer.core.instrument
- io.micrometer.core.instrument.simple
- io.opentelemetry.api
- io.opentelemetry.api.trace
- io.opentelemetry.context
- io.opentelemetry.context.propagation
- jakarta.persistence
- jakarta.validation
- javax.annotation
- javax.cache
- javax.cache.configuration
- javax.cache.event
- javax.persistence
- javax.validation
- net.sf.ehcache
- net.sf.ehcache.event
- nu.xom
- org.axonframework.metrics
- org.axonframework.micrometer
- org.axonframework.spring
- org.axonframework.spring.config
- org.axonframework.spring.config.annotation
- org.axonframework.spring.eventsourcing
- org.axonframework.spring.jdbc
- org.axonframework.spring.messaging.unitofwork
- org.axonframework.spring.saga



Table 7 Chart 2 - Top 15 external dependency using artifacts and their external packages (first 2 levels) stacked

The following chart shows the top 15 external package using artifacts and breaks down which external packages (first 2 levels) are used in how many different internal packages with stacked bars.

Note that every external dependency is counted separately so that if on internal package uses two external packages it will be displayed for both and so stacked twice.

<Figure size 640x480 with 0 Axes>

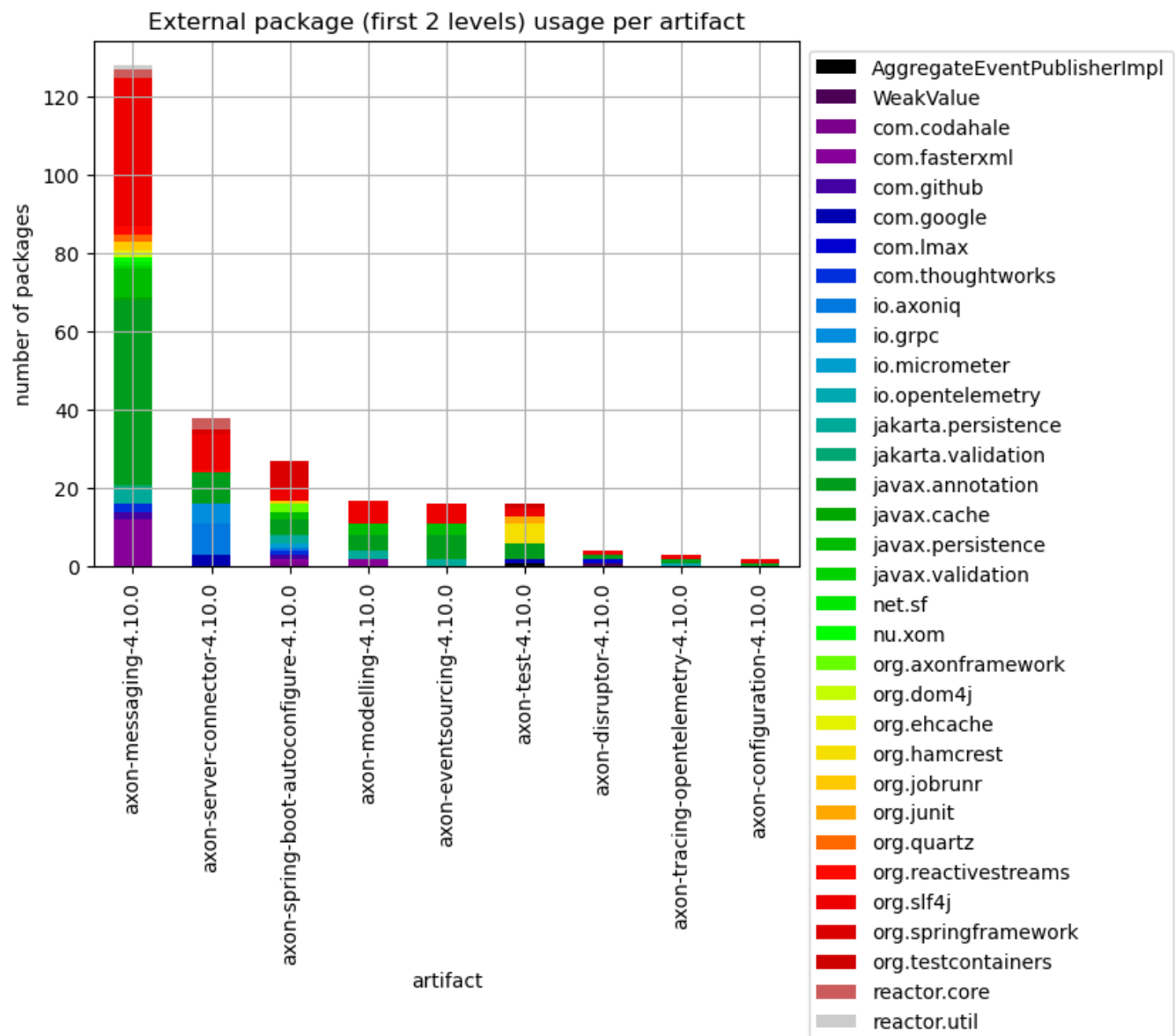


Table 8 - External usage per artifact

The following table shows the most used external packages separately for each artifact including external annotations. The results are grouped per artifact and sorted by the artifacts with the highest external type usage rate descending. Additionally, for each artifact the top 5 used external packages are listed in the top5ExternalPackages column.

The intention of this table is to find artifacts that use a lot of external dependencies in relation to their size and get an overview per artifact with the top 5 used external packages, the number of external types and packages used etc. .

Only the last 40 entries are shown. The whole table can be found in the following CSV report:

External\_package\_usage\_per\_artifact\_sorted\_top

Columns:

- *artifactName* is used to group the the external package usage per artifact for a more detailed analysis.
- *numberOfTypesInArtifact* represents the total count of all analyzed types for the artifact
- *numberOfExternalTypesInArtifact* is the number of all external types that are used by the artifact
- *numberOfExternalPackagesInArtifact* is the number of all external packages that are used by the artifact
- *externalTypeRate* is the  $\text{numberOfExternalTypesInArtifact} / \text{numberOfTypesInArtifact} * 100$
- *numberOfExternalTypeCaller* refers to the distinct types that make use of the external package
- *numberOfExternalTypeCalls* includes every invocation or reference to the types in the external package
- *numberOfExternalPackages* is the number of distinct external packages used by the artifact
- *top5ExternalPackages* contains a list of the top 5 most used external packages of the artifact
- *someExternalTypes* contains a list of lists and is also mean't to provide some examples of external types used

	artifactName	numberOfTypesInArtifact	numberOfExternalTypesInArtifact	numberOfExternalPackagesInArtifact	externalTypeRate	numberOfExternalTypeCaller	nu
0	axon-tracing-opentelemetry-4.10.0	5	16	6	320.000000	26	
1	axon-spring-boot-autoconfigure-4.10.0	74	101	47	136.486486	304	
2	axon-server-connector-4.10.0	131	113	26	86.259542	341	
3	axon-disruptor-4.10.0	22	13	5	59.090909	33	
4	axon-test-4.10.0	87	28	13	32.183908	99	
5	axon-modelling-4.10.0	156	35	5	22.435897	99	
6	axon-messaging-4.10.0	787	154	42	19.567980	747	
7	axon-event sourcing-4.10.0	133	25	4	18.796992	74	
8	axon-configuration-4.10.0	41	4	2	9.756098	24	

**Table 9 - External usage per artifact and package**

This table lists internal packages and the artifacts they belong to that use many different external types of a specific external package without taking external annotations into account.

Only the last 40 entries are shown. The whole table can be found in the following CSV report:

`External_package_usage_per_artifact_and_package`

**Columns:**

- *artifactName* that contains the type that calls the external package
- *fullPackageName* is the package within the artifact that contains the type that calls the external package

- *externalPackageName* identifies the external package as described above
- *numberOfExternalTypeCaller* refers to the distinct types that make use of the external package
- *numberOfExternalTypeCalls* includes every invocation or reference to the types in the external package
- *numberOfTypesInPackage* represents the total count of all types in that package
- *externalTypeNames* contains a list of actually utilized types of the external package
- *packageName* contains the name of the package (last part of *fullPackageName*)

	artifactName	fullPackageName	externalPackageName	numberOfExternalTypeCaller	numberOfExternalTypeCalls	num
0	axon-server-connector-4.10.0	org.axonframework.axonserver.connector.util	io.grpc	39	69	
1	axon-test-4.10.0	org.axonframework.test.matchers	org.hamcrest	38	147	
2	axon-messaging-4.10.0	org.axonframework.queryhandling	reactor.core.publisher	28	108	
3	axon-server-connector-4.10.0	org.axonframework.axonserver.connector.query	io.axoniq.axonserver.grpc.query	25	124	
4	axon-server-connector-4.10.0	org.axonframework.axonserver.connector.query.s...	io.axoniq.axonserver.grpc.query	21	113	
5	axon-server-connector-4.10.0	org.axonframework.axonserver.connector.util	io.axoniq.axonserver.grpc	20	106	
6	axon-messaging-4.10.0	org.axonframework.deadline.quartz	org.quartz	18	109	
7	axon-server-connector-4.10.0	org.axonframework.axonserver.connector.event.axon	io.axoniq.axonserver.connector.event	18	89	
8	axon-messaging-4.10.0	org.axonframework.eventhandling.scheduling.quartz	org.quartz	17	76	
9	axon-server-connector-4.10.0	org.axonframework.axonserver.connector.query	io.axoniq.axonserver.connector	17	53	
10	axon-messaging-4.10.0	org.axonframework.eventhandling	org.slf4j	16	56	
11	axon-messaging-4.10.0	org.axonframework.eventhandling.pooled	org.slf4j	15	71	
12	axon-messaging-4.10.0	org.axonframework.serialization.json	com.fasterxml.jackson.databind	15	65	
13	axon-disruptor-4.10.0	org.axonframework.disruptor.commandhandling	org.slf4j	12	22	
14	axon-configuration-4.10.0	org.axonframework.config	org.slf4j	11	32	
15	axon-disruptor-4.10.0	org.axonframework.disruptor.commandhandling	com.lmax.disruptor	9	25	
16	axon-server-connector-4.10.0	org.axonframework.axonserver.connector.command	io.axoniq.axonserver.grpc.command	9	52	
17	axon-server-connector-4.10.0	org.axonframework.axonserver.connector.query	io.axoniq.axonserver.grpc	9	46	
18	axon-tracing-opentelemetry-4.10.0	org.axonframework.tracing.opentelemetry	io.opentelemetry.context.propagation	9	18	
19	axon-tracing-opentelemetry-4.10.0	org.axonframework.tracing.opentelemetry	io.opentelemetry.api.trace	9	47	
20	axon-eventsourcing-4.10.0	org.axonframework.eventsourcing.eventstore.leg...	org.slf4j	8	15	
21	axon-messaging-4.10.0	org.axonframework.common.caching	org.ehcache.event	8	30	
22	axon-messaging-4.10.0	org.axonframework.common.caching	javax.cache.event	8	26	
23	axon-messaging-4.10.0	org.axonframework.messaging.annotation	org.slf4j	8	15	
24	axon-server-connector-4.10.0	org.axonframework.axonserver.connector	io.axoniq.axonserver.grpc	8	68	
25	axon-server-connector-4.10.0	org.axonframework.axonserver.connector.command	io.axoniq.axonserver.grpc	8	25	
26	axon-server-connector-4.10.0	org.axonframework.axonserver.connector.event.axon	org.slf4j	8	24	

	artifactName	fullPackageName	externalPackageName	numberOfExternalTypeCaller	numberOfExternalTypeCalls	num
27	axon-server-connector-4.10.0	org.axonframework.axonserver.connector.event.axon	io.axoniq.axonserver.grpc.event	8	43	
28	axon-test-4.10.0	org.axonframework.test.saga	org.hamcrest	8	66	
29	axon-messaging-4.10.0	org.axonframework.messaging.responsetypes	reactor.core.publisher	7	34	
30	axon-messaging-4.10.0	org.axonframework.queryhandling	org.reactivestreams	7	23	
31	axon-messaging-4.10.0	org.axonframework.queryhandling	org.slf4j	7	15	
32	axon-server-connector-4.10.0	org.axonframework.axonserver.connector.event.axon	io.axoniq.axonserver.grpc	7	18	
33	axon-server-connector-4.10.0	org.axonframework.axonserver.connector.query	reactor.core.publisher	7	27	
34	axon-server-connector-4.10.0	org.axonframework.axonserver.connector.query.s...	reactor.core.publisher	7	29	
35	axon-spring-boot-autoconfigure-4.10.0	org.axonframework.springboot.autoconfig	org.springframework.context	7	28	
36	axon-eventsourcing-4.10.0	org.axonframework.eventsourcing.eventstore	org.slf4j	6	9	
37	axon-messaging-4.10.0	org.axonframework.eventhandling.deadletter	org.slf4j	6	25	
38	axon-messaging-4.10.0	org.axonframework.messaging.responsetypes	org.reactivestreams	6	14	
39	axon-messaging-4.10.0	org.axonframework.serialization	com.thoughtworks.xstream.converters	6	6	

Table 10 - Top 20 external package usage per type

This table shows internal types that utilize the most different external types and packages. These have the highest probability of change depending on external libraries. A case-by-case approach is also advisable here because there could for example also be code units that encapsulate an external library and have this high count of external dependencies on purpose.

Only the last 20 entries are shown. The whole table can be found in the following CSV report:

[External\\_package\\_usage\\_per\\_type](#)

#### Columns:

- *artifactName* that contains the type that calls the external package
- *fullPackageName* is the package within the artifact that contains the type that calls external types
- *typeName* identifies the internal type within the package and artifact that calls external types
- *numberOfExternalTypeCaller* and *numberOfExternalTypes* refers to the distinct external types that are used by the internal type
- *numberOfExternalTypeCalls* includes every invocation or reference to the types in the external package
- *numberOfTypesInPackage* represents the total count of all types in that package
- *numberOfExternalPackages* shows how many different external packages are used by the internal type



- *externalPackageNames* contains the list of names of the different external packages that are used by the internal type
- *externalTypeNames* contains a list of actually utilized types of the external package
- *packageName* contains the name of the package (last part of *fullPackageName*)

	artifactName	fullPackageName	typeName	numberOfExternalTypeCaller	numberOfExternalTyp
0	axon-spring-boot-autoconfigure-4.10.0	org.axonframework.springboot.autoconfigure	AxonAutoConfiguration	19	
1	axon-spring-boot-autoconfigure-4.10.0	org.axonframework.springboot.autoconfigure	InfraConfiguration	18	
2	axon-server-connector-4.10.0	org.axonframework.axonserver.connector.query	AxonServerQueryBus	15	
3	axon-spring-boot-autoconfigure-4.10.0	org.axonframework.springboot.autoconfigure	AxonServerAutoConfiguration	12	
4	axon-spring-boot-autoconfigure-4.10.0	org.axonframework.springboot.util	AbstractQualifiedBeanCondition	12	
5	axon-server-connector-4.10.0	org.axonframework.axonserver.connector.event.axon	AxonServerEventStore\$AxonIQEventStorageEngine	12	
6	axon-spring-boot-autoconfigure-4.10.0	org.axonframework.springboot.autoconfigure	XStreamAutoConfiguration	11	
7	axon-spring-boot-autoconfigure-4.10.0	org.axonframework.springboot.autoconfigure	AxonDbSchedulerAutoConfiguration	9	
8	axon-spring-boot-autoconfigure-4.10.0	org.axonframework.springboot.autoconfigure	PersistentStreamMessageSourceRegistrar	12	
9	axon-spring-boot-autoconfigure-4.10.0	org.axonframework.springboot.autoconfigure	MicrometerMetricsAutoConfiguration	13	
10	axon-spring-boot-autoconfigure-4.10.0	org.axonframework.springboot.autoconfigure	JpaAutoConfiguration	10	
11	axon-spring-boot-autoconfigure-4.10.0	org.axonframework.springboot.autoconfigure.legacyjpa	JpaJavaxAutoConfiguration	11	
12	axon-messaging-4.10.0	org.axonframework.serialization.json	JacksonSerializer	9	
13	axon-server-connector-4.10.0	org.axonframework.axonserver.connector	AxonServerConnectionManager\$Builder	7	
14	axon-server-connector-4.10.0	org.axonframework.axonserver.connector.event.axon	AxonServerEventScheduler	10	
15	axon-server-connector-4.10.0	org.axonframework.axonserver.connector.processor	EventProcessorControlService	8	
16	axon-server-connector-4.10.0	org.axonframework.axonserver.connector.query	QueryProcessingTask	10	
17	axon-spring-boot-autoconfigure-4.10.0	org.axonframework.springboot.autoconfigure	JpaEventStoreAutoConfiguration	9	
18	axon-spring-boot-autoconfigure-4.10.0	org.axonframework.springboot.autoconfigure	MetricsAutoConfiguration	11	
19	axon-spring-boot-autoconfigure-4.10.0	org.axonframework.springboot.autoconfigure	JdbcAutoConfiguration	9	

Table 11 - External package usage distribution per type

This table shows how many types use one external package, how many use two, etc. . This gives an overview of the distribution of external package calls and the overall coupling to external libraries. The higher the count of distinct external packages the lower should be the count of types that use them. Dependencies to external annotations are left out here.

More details about which types have the highest external package dependency usage can be in the tables 4 and 5 above.

Only the last 40 entries are shown. The whole table can be found in the following CSV report:

External\_package\_usage\_per\_artifact\_distribution

Columns:

- *artifactName* that contains the type that calls the external package
- *artifactTypes* the total count of types in the artifact
- *numberOfExternalPackages* the number of distinct external packages used
- *numberOfTypes* in the artifact where the *numberOfExternalPackages* applies
- *numberOfTypesPercentage* in the artifact where the *numberOfExternalPackages* applies in %

	artifactName	artifactPackages	artifactTypes	numberOfExternalPackages	numberOfPackages	numberOfTypes	typesCallingExternalRate	packagesCallingExternalRate
0	axon-messaging-4.10.0	64	787	41	44	155	19.695044	68.71
1	axon-server-connector-4.10.0	11	131	24	10	82	62.595420	90.91
2	axon-spring-boot-autoconfigure-4.10.0	9	74	43	8	35	47.297297	88.81
3	axon-modelling-4.10.0	10	156	3	7	12	7.692308	70.00
4	axon-test-4.10.0	8	87	12	6	36	41.379310	75.00
5	axon-event sourcing-4.10.0	9	133	3	5	15	11.278195	55.56
6	axon-configuration-4.10.0	1	41	1	1	6	14.634146	100.00
7	axon-disruptor-4.10.0	1	22	4	1	10	45.454545	100.00
8	axon-tracing-opentelemetry-4.10.0	1	5	5	1	5	100.000000	100.00

Table 12 - External package usage per artifact grouped by number of internal packages

The following table shows the external package usage for every artifact grouped by the number of distinct internal dependent packages. The intention is to find external package usage spread across multiple internal packages in artifacts.

Artifacts that encapsulate external dependency calls in one internal package overall (or each) are easier to change if those external dependencies change and are most likely applying a [Hexagonal architecture](#). Artifacts that use external dependencies in multiple internal packages need more effort to adapt to changes of those external dependencies. On one hand this could be intended e.g. when using standardized libraries. On the other hand this might indicate higher than necessary coupling.

The whole table can be found in the following CSV report:

External\_package\_usage\_per\_internal\_package\_count

artifactName	axon-eventsourcing-4.10.0	axon-messaging-4.10.0	axon-modelling-4.10.0	axon-server-connector-4.10.0	axon-spring-boot-autoconfigure-4.10.0	axon-test-4.10.0
numberOfPackages						
2	22.222222	3.1250	20.0	18.181818	22.222222	25.0
3	33.333333	0.0000	30.0	27.272727	33.333333	0.0
4	0.000000	6.2500	40.0	36.363636	44.444444	50.0
5	55.555556	0.0000	0.0	0.000000	0.000000	62.5
6	66.666667	0.0000	60.0	54.545455	0.000000	0.0
7	0.000000	10.9375	0.0	63.636364	0.000000	0.0
8	0.000000	0.0000	0.0	72.727273	0.000000	0.0
9	0.000000	0.0000	0.0	81.818182	0.000000	0.0
11	0.000000	17.1875	0.0	0.000000	0.000000	0.0
38	0.000000	59.3750	0.0	0.000000	0.000000	0.0
47	0.000000	73.4375	0.0	0.000000	0.000000	0.0

## Table 13 - External package usage aggregated

This table lists all artifacts and their external package dependencies usage aggregated over internal packages.

The intention behind this is to find artifacts that use an external dependency across multiple internal packages. This might be intended for frameworks and standardized libraries and helps to quantify how widely those are used. For some external dependencies it might be beneficial to only access it from one package and provide an abstraction for internal usage following a [Hexagonal architecture](#). Thus, this table may also help in finding application for the Hexagonal architecture or similar approaches (Domain Driven Design Anti Corruption Layer). After all it is easier to update or replace such external dependencies when they are used in specific areas and not all over the code.

Only the last 40 entries are shown. The whole table can be found in the following CSV report:

External\_package\_usage\_per\_artifact\_package\_aggregated

### Columns:

- *artifactName* that contains the type that calls the external package
- *artifactPackages* is the total count of packages in the artifact
- *artifactTypes* is the total count of types in the artifact
- *numberOfExternalPackages* the number of distinct external packages used
- *[min,max,med,avg,std]NumberOfPackages* provide statistics based on each external package and its package usage within the artifact

- *[min,max,med,avg,std]NumberOfPackagesPercentage* provide statistics in % based on each external package and its package usage within the artifact in respect to the overall count of packages in the artifact
- *[min,max,med,avg,std]NumberOfTypes* provide statistics based on each external package and its type usage within the artifact
- *[min,max,med,avg,std]NumberOfTypePercentage* provide statistics in % based on each external package and its type usage within the artifact in respect to the overall count of packages in the artifact
- *numberOfTypes* in the artifact where the *numberOfExternalPackages* applies
- *numberOfTypesPercentage* in the artifact where the *numberOfExternalPackages* applies in %

Table 13a - External package usage aggregated - count of internal packages

	artifactName	artifactPackages	numberOfExternalPackages	minNumberOfPackages	medNumberOfPackages	avgNumberOfPackages	maxNumberOfPackages	std
0	axon-messaging-4.10.0	64	41	1	1.0	2.292683	38	
1	axon-server-connector-4.10.0	11	24	1	2.0	2.500000	9	
2	axon-modelling-4.10.0	10	3	2	3.0	3.666667	6	
3	axon-event sourcing-4.10.0	9	3	1	2.0	2.666667	5	
4	axon-test-4.10.0	8	12	1	1.0	1.416667	5	
5	axon-spring-boot-autoconfigure-4.10.0	9	43	1	1.0	1.279070	3	
6	axon-configuration-4.10.0	1	1	1	1.0	1.000000	1	
7	axon-disruptor-4.10.0	1	4	1	1.0	1.000000	1	
8	axon-tracing-opentelemetry-4.10.0	1	5	1	1.0	1.000000	1	

Table 13b - External package usage aggregated - percentage of internal packages

	artifactName	artifactPackages	numberOfExternalPackages	minNumberOfPackagesPercentage	medNumberOfPackagesPercentage	avgNumberOfPackagesPercentage
0	axon-messaging-4.10.0	64	41	1.562500	1.562500	3.5823
1	axon-server-connector-4.10.0	11	24	9.090909	18.181818	22.7273
2	axon-modelling-4.10.0	10	3	20.000000	30.000000	36.6667
3	axon-eventsourcing-4.10.0	9	3	11.111111	22.222222	29.6296
4	axon-test-4.10.0	8	12	12.500000	12.500000	17.7083
5	axon-spring-boot-autoconfigure-4.10.0	9	43	11.111111	11.111111	14.2118
6	axon-configuration-4.10.0	1	1	100.000000	100.000000	100.0000
7	axon-disruptor-4.10.0	1	4	100.000000	100.000000	100.0000
8	axon-tracing-opentelemetry-4.10.0	1	5	100.000000	100.000000	100.0000

Table 13c - External package usage aggregated - count of internal types

	artifactName	artifactTypes	numberOfExternalPackages	minNumberOfTypes	medNumberOfTypes	avgNumberOfTypes	maxNumberOfTypes	stdNumberOfTypes
0	axon-messaging-4.10.0	787	41	1	3.0	5.390244	78	12.122042
1	axon-server-connector-4.10.0	131	24	1	4.0	7.125000	30	7.719611
2	axon-modelling-4.10.0	156	3	3	3.0	5.000000	9	3.464102
3	axon-eventsourcing-4.10.0	133	3	3	3.0	6.666667	14	6.350853
4	axon-test-4.10.0	87	12	1	1.5	3.833333	27	7.346407
5	axon-spring-boot-autoconfigure-4.10.0	74	43	1	1.0	1.883721	6	1.159008
6	axon-configuration-4.10.0	41	1	6	6.0	6.000000	6	0.000000
7	axon-disruptor-4.10.0	22	4	1	5.5	5.000000	8	3.162278
8	axon-tracing-opentelemetry-4.10.0	5	5	1	2.0	2.200000	4	1.303840

Table 13d - External package usage aggregated - percentage of internal types

	artifactName	artifactTypes	numberOfExternalPackages	minNumberOfTypesPercentage	medNumberOfTypesPercentage	avgNumberOfTypesPercentage	maxNumb
0	axon-messaging-4.10.0	787	41	0.127065	0.381194	0.684910	
1	axon-server-connector-4.10.0	131	24	0.763359	3.053435	5.438931	
2	axon-modelling-4.10.0	156	3	1.923077	1.923077	3.205128	
3	axon-eventsourcing-4.10.0	133	3	2.255639	2.255639	5.012531	
4	axon-test-4.10.0	87	12	1.149425	1.724138	4.406130	
5	axon-spring-boot-autoconfigure-4.10.0	74	43	1.351351	1.351351	2.545569	
6	axon-configuration-4.10.0	41	1	14.634146	14.634146	14.634146	
7	axon-disruptor-4.10.0	22	4	4.545455	25.000000	22.727273	
8	axon-tracing-opentelemetry-4.10.0	5	5	20.000000	40.000000	44.000000	

Table 13 Chart 1 - External package usage - max percentage of internal types

This chart shows per artifact the maximum percentage of internal packages (compared to all packages in that artifact) that use one specific external package.

**Example:** One artifact might use 10 external packages where 7 of them are used in one internal package, 2 of them are used in two packages and one external dependency is used in 5 packages. So for this artifact there will be a point at x = 10 (external packages used by the artifact) and 5 (max internal packages). Instead of the count the percentage of internal packages compared to all packages in that artifact is used to get a normalized plot.

<Figure size 640x480 with 0 Axes>

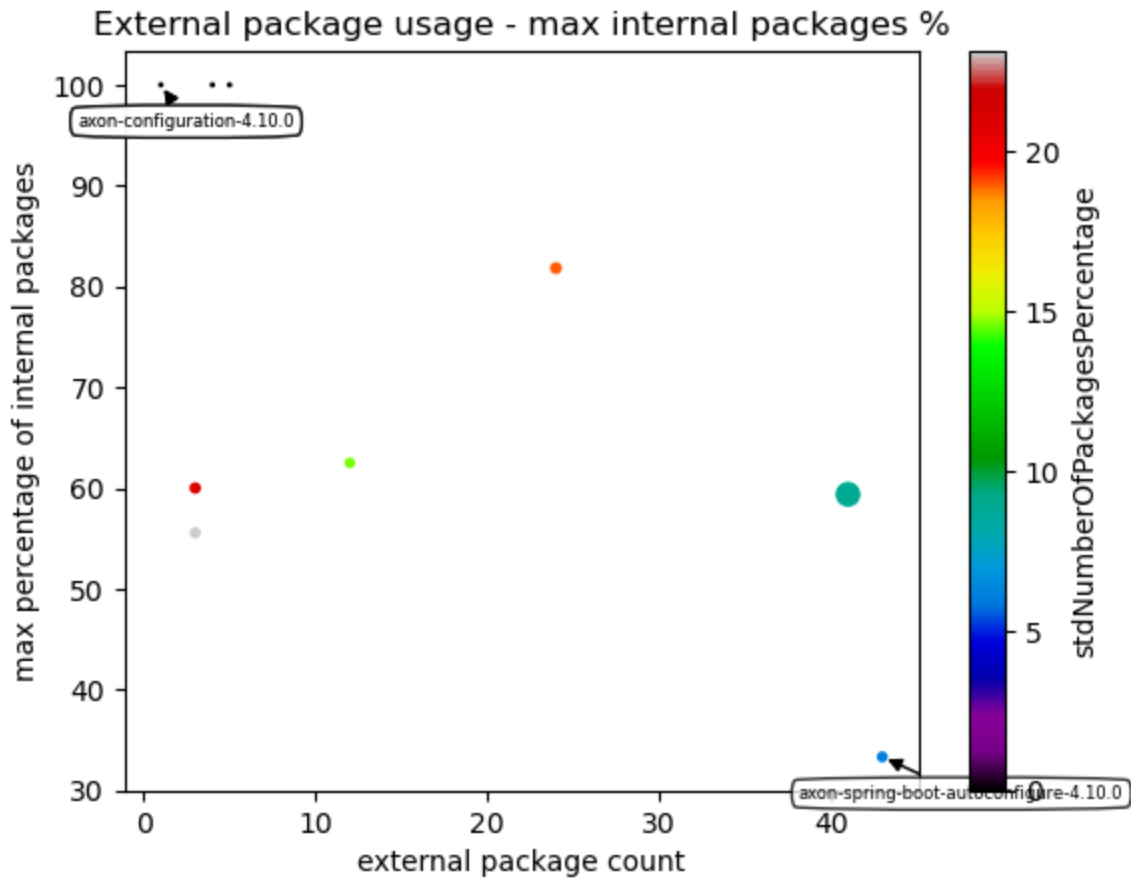


Table 13 Chart 2 - External package usage - median percentage of internal types

This chart shows per artifact the median (0.5 percentile) of internal packages (compared to all packages in that artifact) that use one specific external package.

**Example:** One artifact might use 9 external packages where 3 of them are used in 1 internal package, 3 of them are used in 2 package and the last 3 ones are used in 3 packages. So for this artifact there will be a point at x = 10 (external packages used by the artifact) and 2 (median internal packages). Instead of the count the percentage of internal packages compared to all packages in that artifact is used to get a normalized plot.

<Figure size 640x480 with 0 Axes>



## Maven POMs

Table 14 - Maven POMs and their declared dependencies

If Maven is used as for package and dependency management and a ".pom" file is included in the artifact, the following table shows the external dependencies that are declared there.

	pom.artifactId	pom.name	scope	dependency.optional	dependentArtifact.group	dependentArtifact.name
0	axon-configuration	pom	test	False	org.hibernate	hibernate-core-jakarta
1	axon-configuration	pom	default	False	org.axonframework	axon-messaging
2	axon-configuration	pom	test	False	org.quartz-scheduler	quartz
3	axon-configuration	pom	default	False	\$(project.groupId)	axon-disruptor
4	axon-configuration	pom	test	False	org.hsqldb	hsqldb
...	...	...	...	...	...	...
158	axon-test	pom	default	True	com.google.code.gson	gson
159	axon-tracing-opentelemetry	pom	provided	False	com.google.code.findbugs	jsr305
160	axon-tracing-opentelemetry	pom	default	False	\$(project.groupId)	axon-messaging
161	axon-tracing-opentelemetry	pom	default	False	\$(project.groupId)	axon-configuration
162	axon-tracing-opentelemetry	pom	default	False	io.opentelemetry	opentelemetry-api

163 rows × 6 columns