

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	n
0	javax.annotation	78	353	386	
1	org.slf4j	67	151	267	
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	15	62	
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	28	61	
16	org.springframework.boot.autoconfigure.condition	3	41	75	
17	org.springframework.context.annotation	3	36	48	
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)

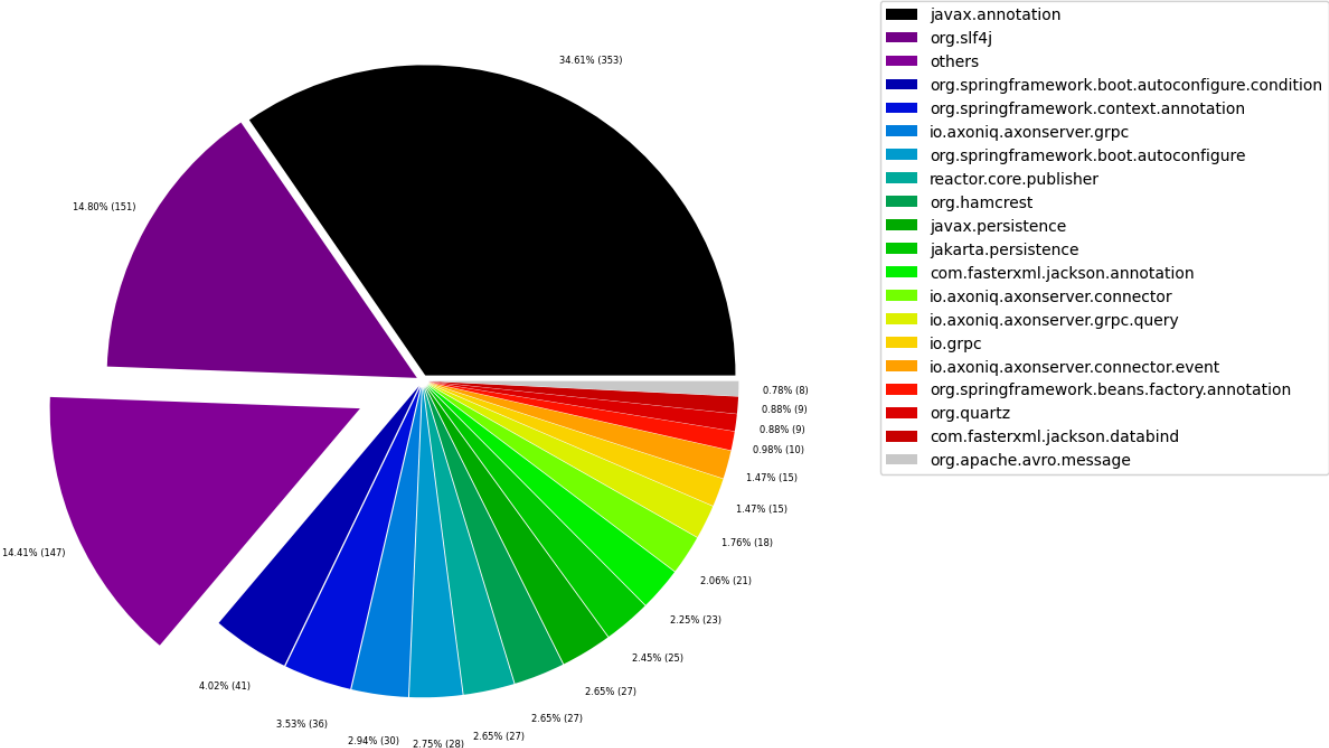


Table 1 Chart 1b - Most called external packages in % by types (less than 0.7% overall "others" drill-down)

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)

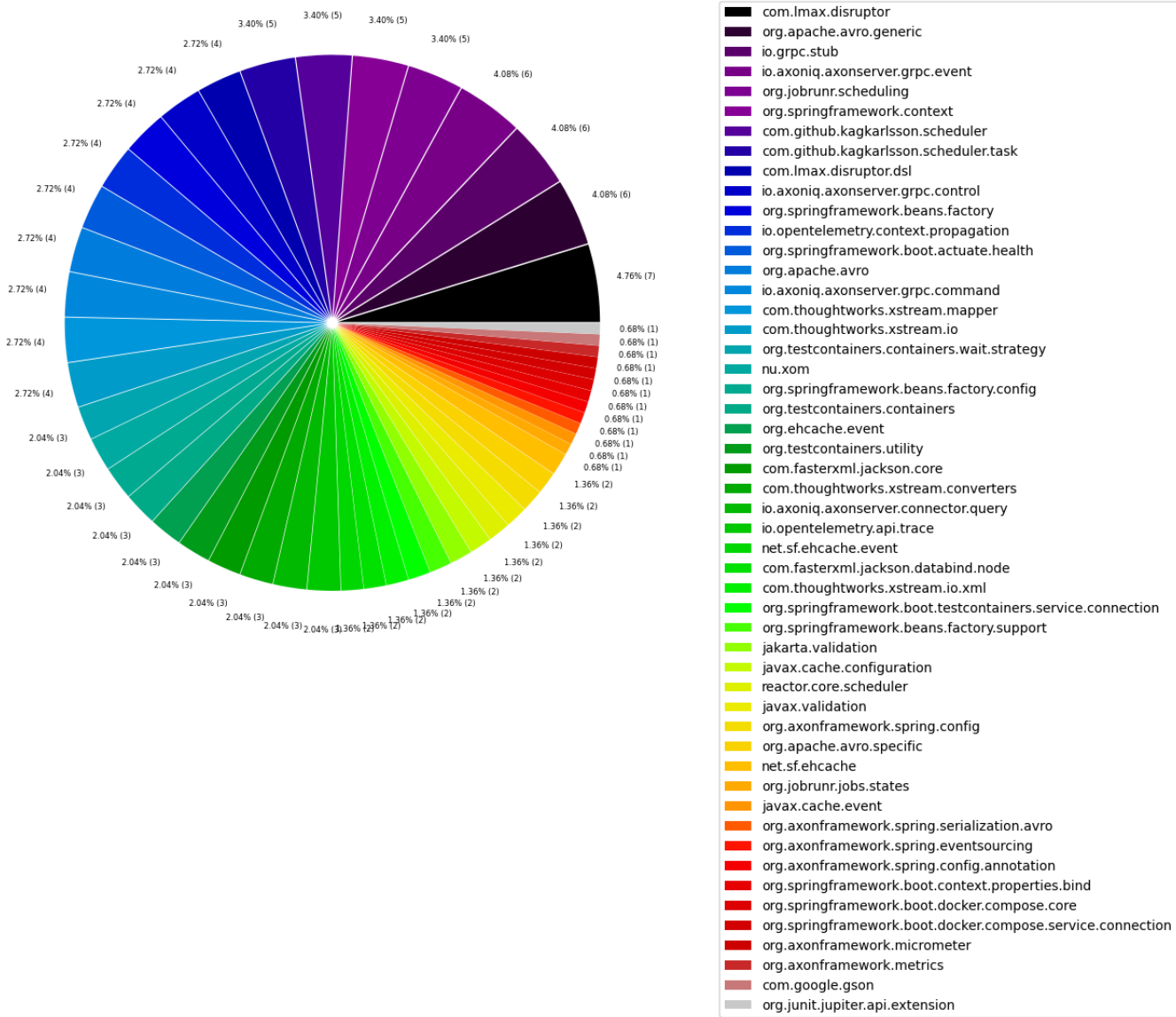


Table 1 Chart 2a - Most called external packages in % by packages (more than 0.7% overall)

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)

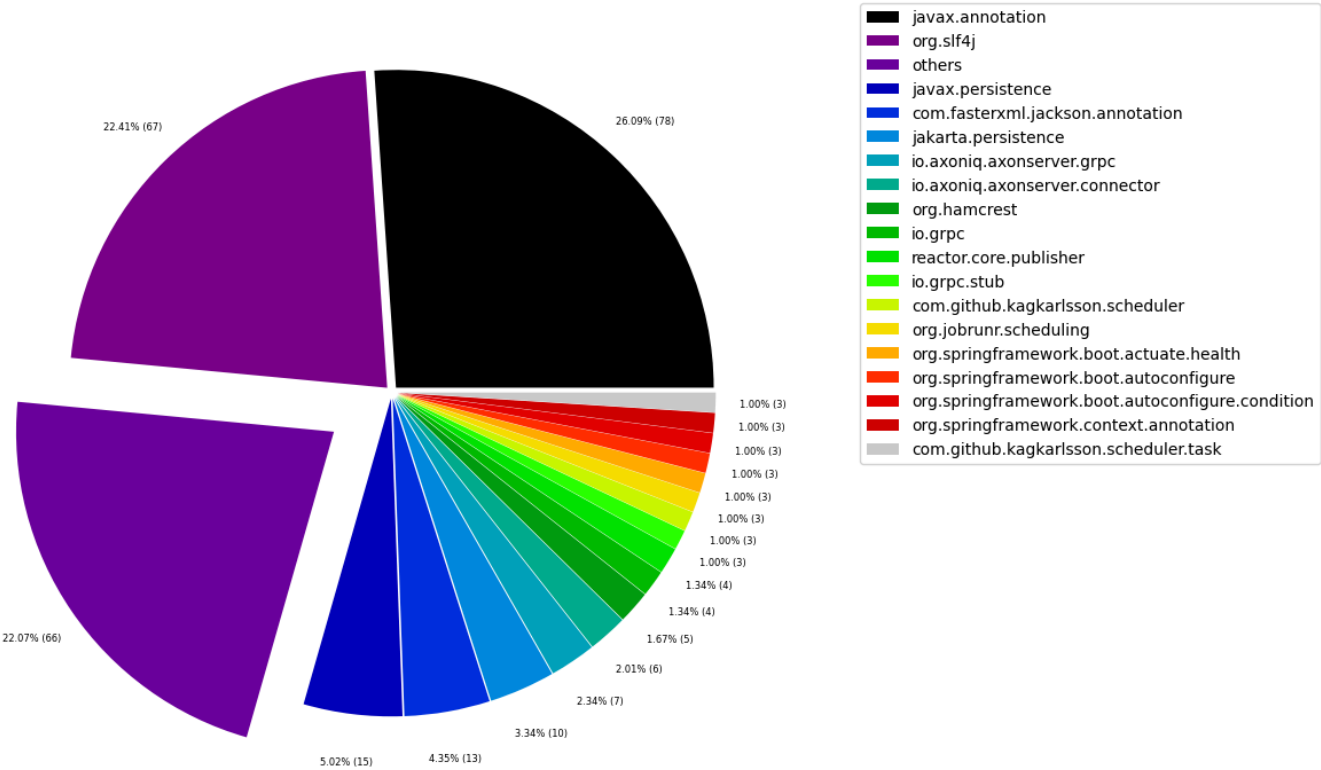


Table 1 Chart 2b - Most called external packages in % by packages (less than 0.7% overall "others" drill-down)

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)

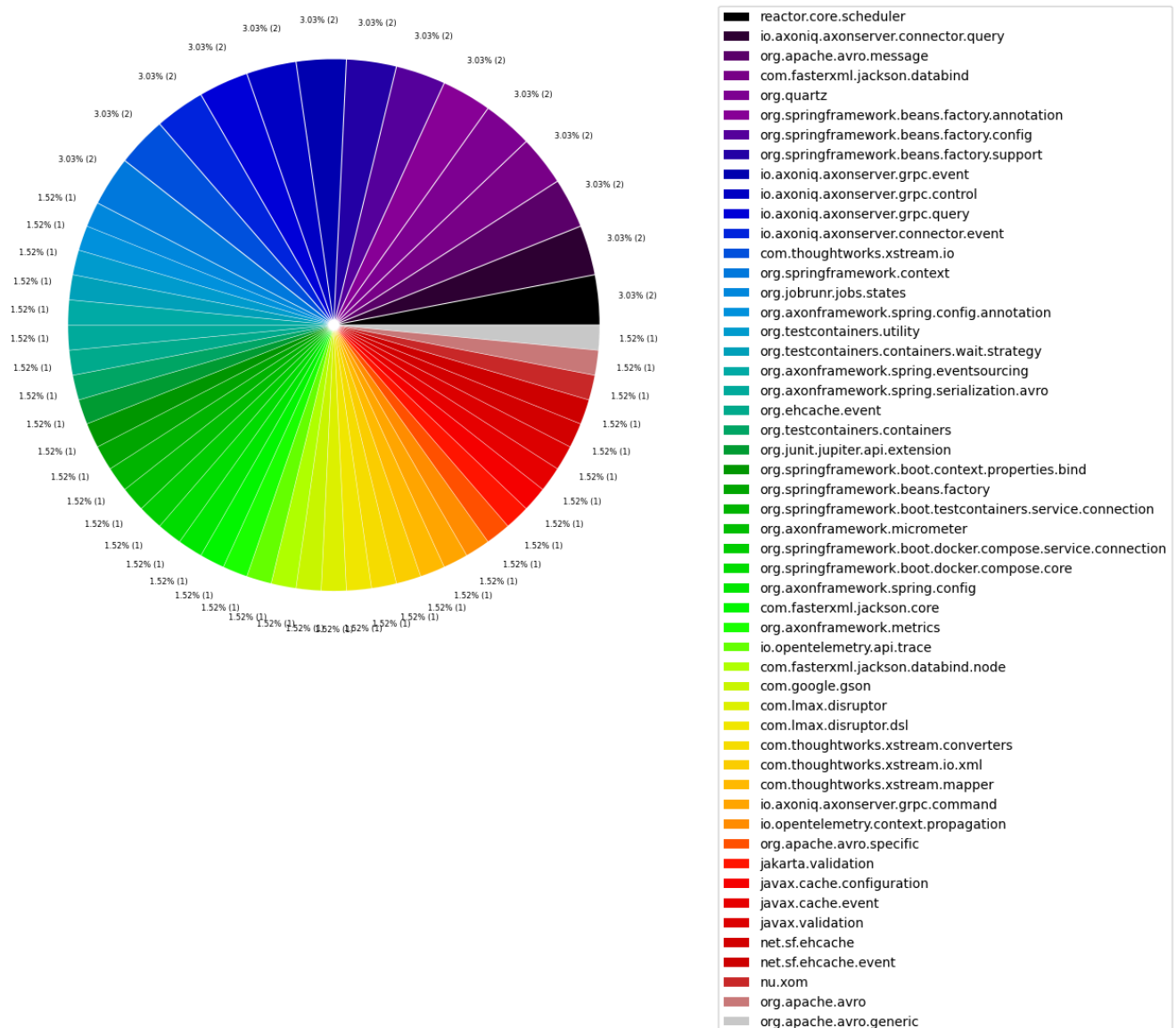


Table 2 - Top 20 most used external packages grouped by their first 2 layers

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	numberOfE
0	javax.annotation	78	353	386	
1	org.slf4j	67	151	267	
2	com.fasterxml	15	33	89	
3	javax.persistence	15	27	82	
4	jakarta.persistence	10	25	73	
5	io.axoniq	9	69	202	
6	org.springframework	8	65	276	
7	io.grpc	5	21	73	
8	org.hamcrest	5	27	59	
9	reactor.core	5	28	53	
10	com.google	4	11	14	
11	com.github	3	7	21	
12	com.thoughtworks	3	10	33	
13	org.jobrunr	3	5	9	
14	org.apache	2	12	41	
15	org.axonframework	2	10	22	
16	org.junit	2	4	8	
17	org.quartz	2	9	36	
18	com.lmax	1	7	14	
19	io.micrometer	1	1	2	

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)

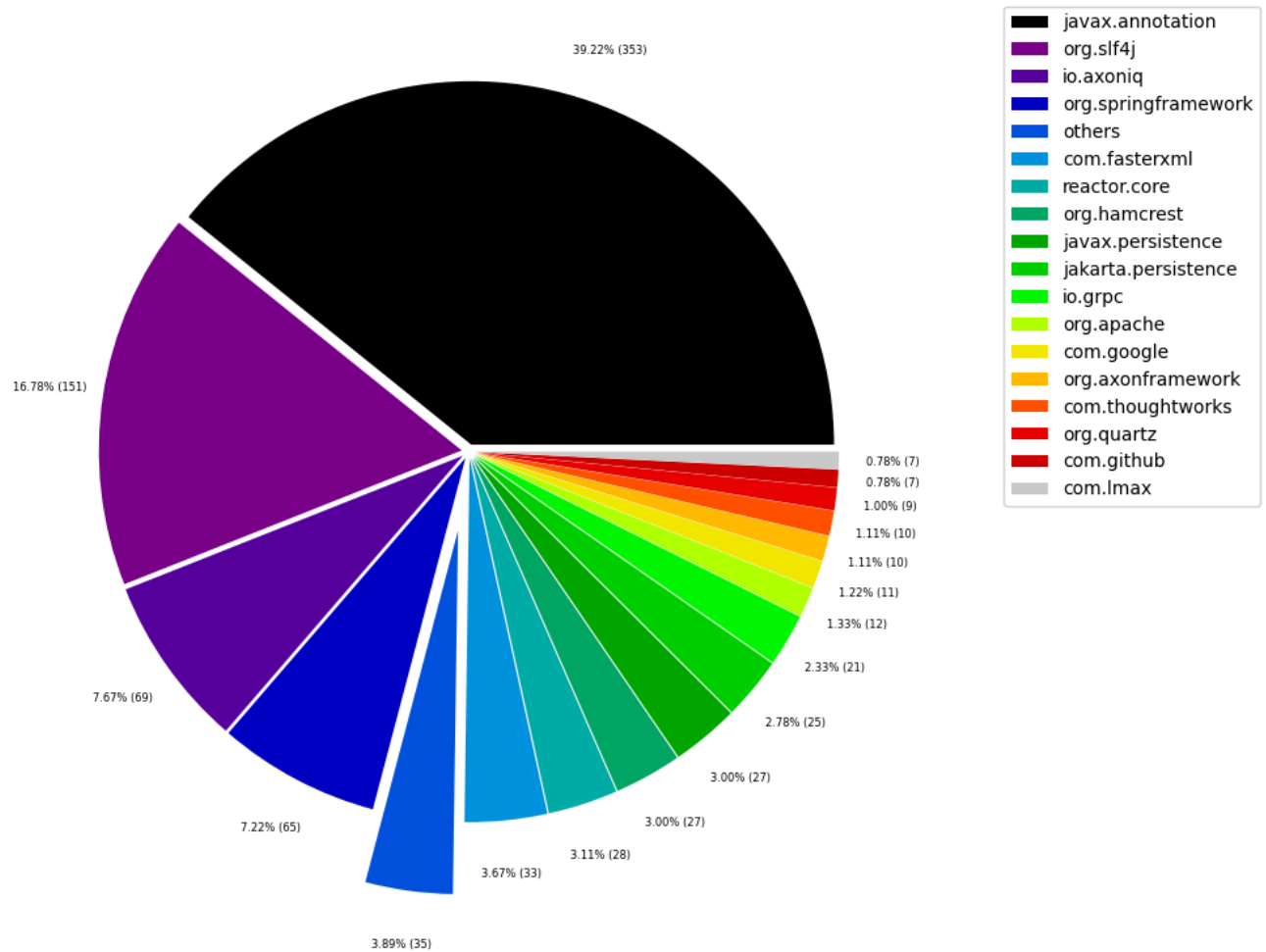


Table 2 Chart 1b - Most called second level external packages in % by type (less than 0.7% overall "others" drill-down)

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)

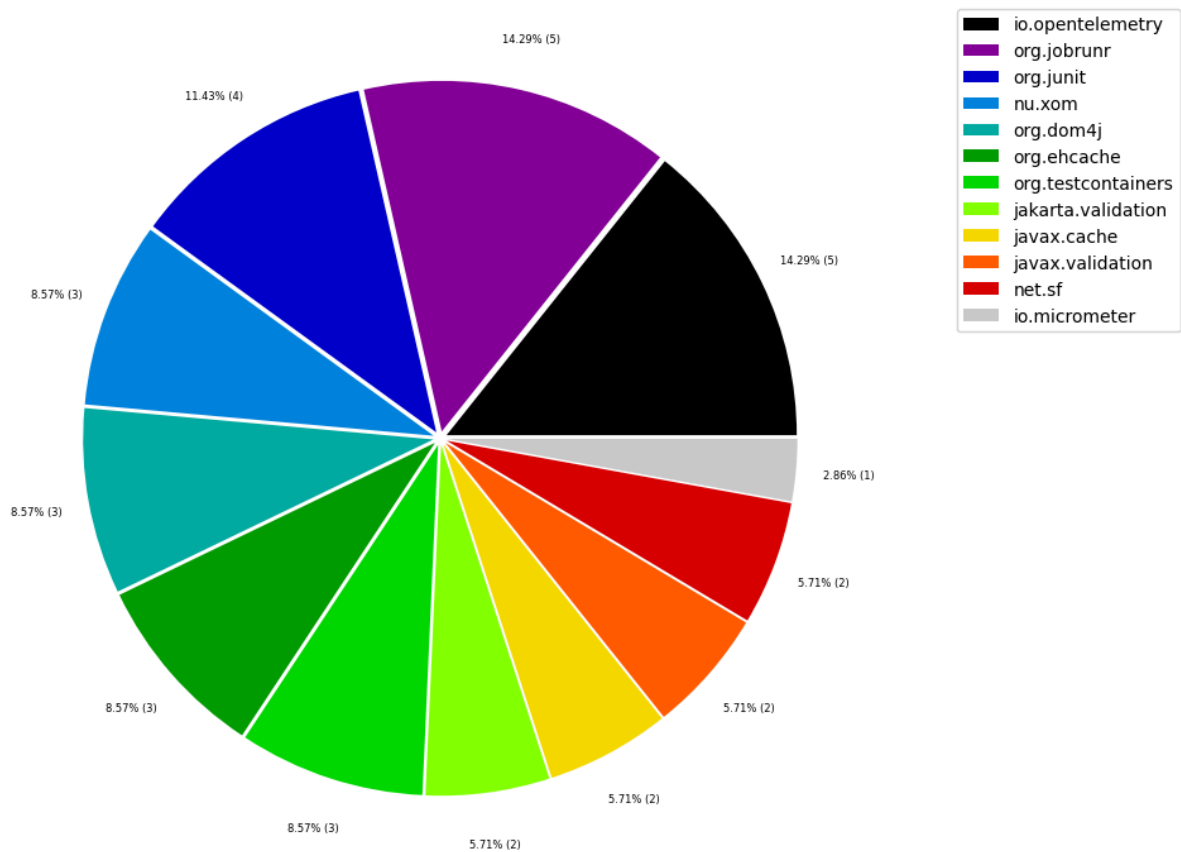


Table 2 Chart 2a - Most called second level external packages in % by package (more than 0.7% overall)

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)

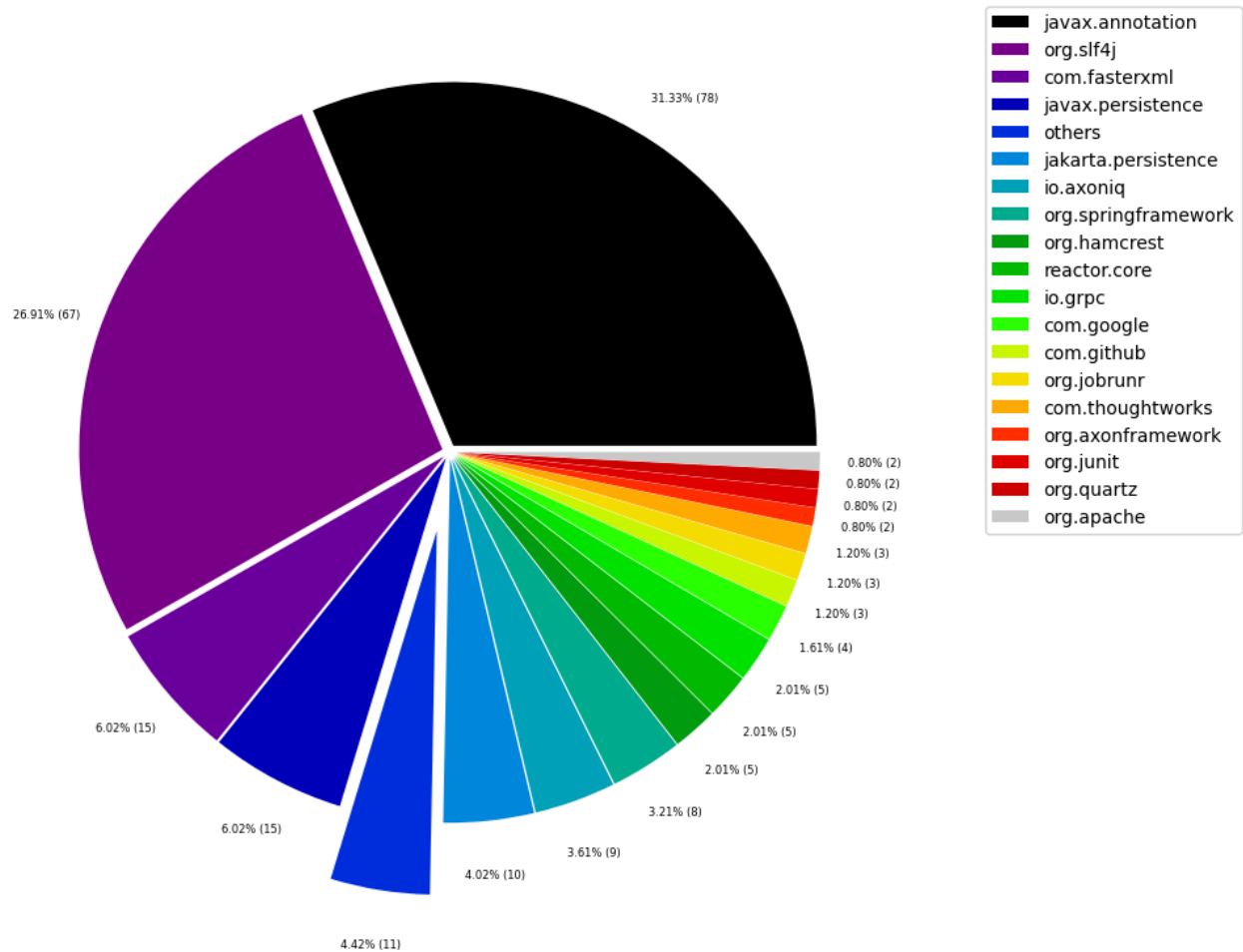


Table 2 Chart 2b - Most called second level external packages in % by package (less than 0.7% overall "others" drill-down)

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)

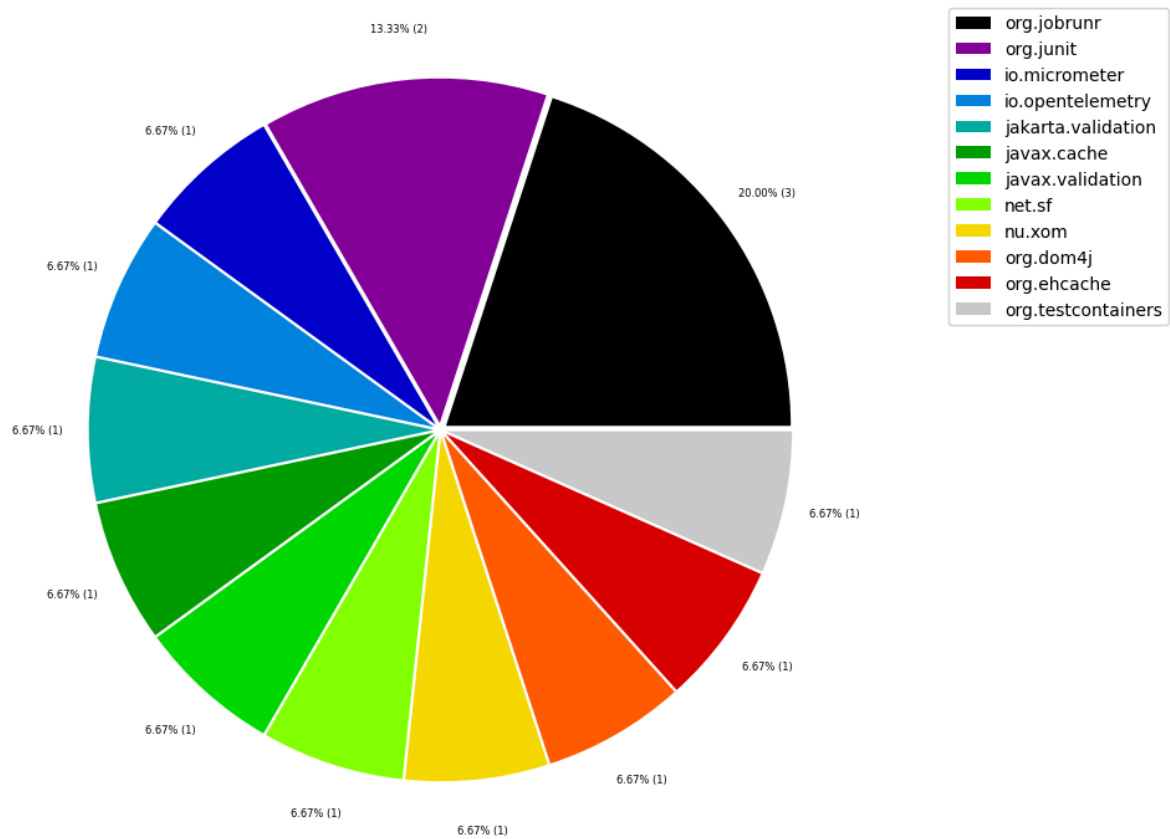


Table 3 - Top 20 most widely spread external packages

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
0	org.slf4j	9	67	1	39	
1	jakarta.persistence	4	8	1	3	
2	javax.persistence	4	11	2	4	
3	com.fasterxml.jackson.databind	2	2	1	1	
4	com.github.kagkarlsson.scheduler	2	3	1	2	
5	com.github.kagkarlsson.scheduler.task	2	3	1	2	
6	com.thoughtworks.xstream	2	3	1	2	
7	io.axoniq.axonserver.connector.event	2	2	1	1	
8	io.axoniq.axonserver.connector.impl	2	4	1	3	
9	org.apache.avro.message	2	2	1	1	

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	avgNumB
0	org.slf4j	9	1	39	3.0	
1	jakarta.persistence	4	1	3	2.0	
2	javax.persistence	4	2	4	2.5	
3	com.fasterxml.jackson.databind	2	1	1	1.0	
4	com.github.kagkarlsson.scheduler	2	1	2	1.5	
5	com.github.kagkarlsson.scheduler.task	2	1	2	1.5	
6	com.thoughtworks.xstream	2	1	2	1.5	
7	io.axoniq.axonserver.connector.event	2	1	1	1.0	
8	io.axoniq.axonserver.connector.impl	2	1	3	2.0	
9	org.apache.avro.message	2	1	1	1.0	
10	org.jobrunr.scheduling	2	1	2	1.5	
11	org.reactivestreams	2	1	2	1.5	
12	reactor.core.publisher	2	2	2	2.0	
13	AggregateEventPublisherImpl	1	1	1	1.0	
14	WeakValue	1	1	1	1.0	
15	com.codahale.metrics	1	1	1	1.0	
16	com.fasterxml.jackson.annotation	1	3	3	3.0	
17	com.fasterxml.jackson.core	1	1	1	1.0	
18	com.fasterxml.jackson.databind.jsontype	1	1	1	1.0	
19	com.fasterxml.jackson.databind.module	1	1	1	1.0	

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	medNumber
0	org.slf4j	9	25.000000	100.000000	
1	jakarta.persistence	4	4.545455	22.222222	
2	javax.persistence	4	6.060606	30.000000	
3	com.fasterxml.jackson.databind	2	1.515152	11.111111	
4	com.github.kagkarlsson.scheduler	2	3.030303	11.111111	
5	com.github.kagkarlsson.scheduler.task	2	3.030303	11.111111	
6	com.thoughtworks.xstream	2	3.030303	11.111111	
7	io.axoniq.axonserver.connector.event	2	9.090909	11.111111	
8	io.axoniq.axonserver.connector.impl	2	11.111111	27.272727	
9	org.apache.avro.message	2	1.515152	11.111111	
10	org.jobrunr.scheduling	2	3.030303	11.111111	
11	org.reactivestreams	2	3.030303	9.090909	
12	reactor.core.publisher	2	3.030303	18.181818	
13	AggregateEventPublisherImpl	1	12.500000	12.500000	
14	WeakValue	1	100.000000	100.000000	
15	com.codahale.metrics	1	11.111111	11.111111	
16	com.fasterxml.jackson.annotation	1	4.545455	4.545455	
17	com.fasterxml.jackson.core	1	1.515152	1.515152	
18	com.fasterxml.jackson.databind.jsontype	1	1.515152	1.515152	
19	com.fasterxml.jackson.databind.module	1	1.515152	1.515152	

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
0	org.slf4j	9	1	82	8.0	16.777778
1	jakarta.persistence	4	3	8	3.0	4.250000
2	javax.persistence	4	3	8	3.0	4.250000
3	com.fasterxml.jackson.databind	2	2	7	4.5	4.500000
4	com.github.kagkarlsson.scheduler	2	1	4	2.5	2.500000
5	com.github.kagkarlsson.scheduler.task	2	1	4	2.5	2.500000
6	com.thoughtworks.xstream	2	2	4	3.0	3.000000
7	io.axoniq.axonserver.connector.event	2	1	14	7.5	7.500000
8	io.axoniq.axonserver.connector.impl	2	1	4	2.5	2.500000
9	org.apache.avro.message	2	3	5	4.0	4.000000
10	org.jobrunr.scheduling	2	1	4	2.5	2.500000
11	org.reactivestreams	2	4	13	8.5	8.500000
12	reactor.core.publisher	2	9	18	13.5	13.500000
13	AggregateEventPublisherImpl	1	1	1	1.0	1.000000
14	WeakValue	1	1	1	1.0	1.000000
15	com.codahale.metrics	1	1	1	1.0	1.000000
16	com.fasterxml.jackson.annotation	1	5	5	5.0	5.000000
17	com.fasterxml.jackson.core	1	3	3	3.0	3.000000
18	com.fasterxml.jackson.databind.jsontype	1	1	1	1.0	1.000000
19	com.fasterxml.jackson.databind.module	1	1	1	1.0	1.000000

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
0	org.slf4j	9	2.298851	36.363636	
1	jakarta.persistence	4	0.988875	3.409091	
2	javax.persistence	4	0.988875	3.409091	
3	com.fasterxml.jackson.databind	2	0.865266	2.272727	
4	com.github.kagkarlsson.scheduler	2	0.494438	1.136364	
5	com.github.kagkarlsson.scheduler.task	2	0.494438	1.136364	
6	com.thoughtworks.xstream	2	0.494438	2.272727	
7	io.axoniq.axonserver.connector.event	2	1.136364	9.859155	
8	io.axoniq.axonserver.connector.impl	2	1.136364	2.816901	
9	org.apache.avro.message	2	0.618047	3.409091	
10	org.jobrunr.scheduling	2	0.494438	1.136364	
11	org.reactivestreams	2	1.606922	2.816901	
12	reactor.core.publisher	2	2.224969	6.338028	
13	AggregateEventPublisherImpl	1	1.149425	1.149425	
14	WeakValue	1	4.545455	4.545455	
15	com.codahale.metrics	1	1.136364	1.136364	
16	com.fasterxml.jackson.annotation	1	0.618047	0.618047	
17	com.fasterxml.jackson.core	1	0.370828	0.370828	
18	com.fasterxml.jackson.databind.jsontype	1	0.123609	0.123609	
19	com.fasterxml.jackson.databind.module	1	0.123609	0.123609	

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>

Top external package usage spread [%] by type (more than 0.5% overall)

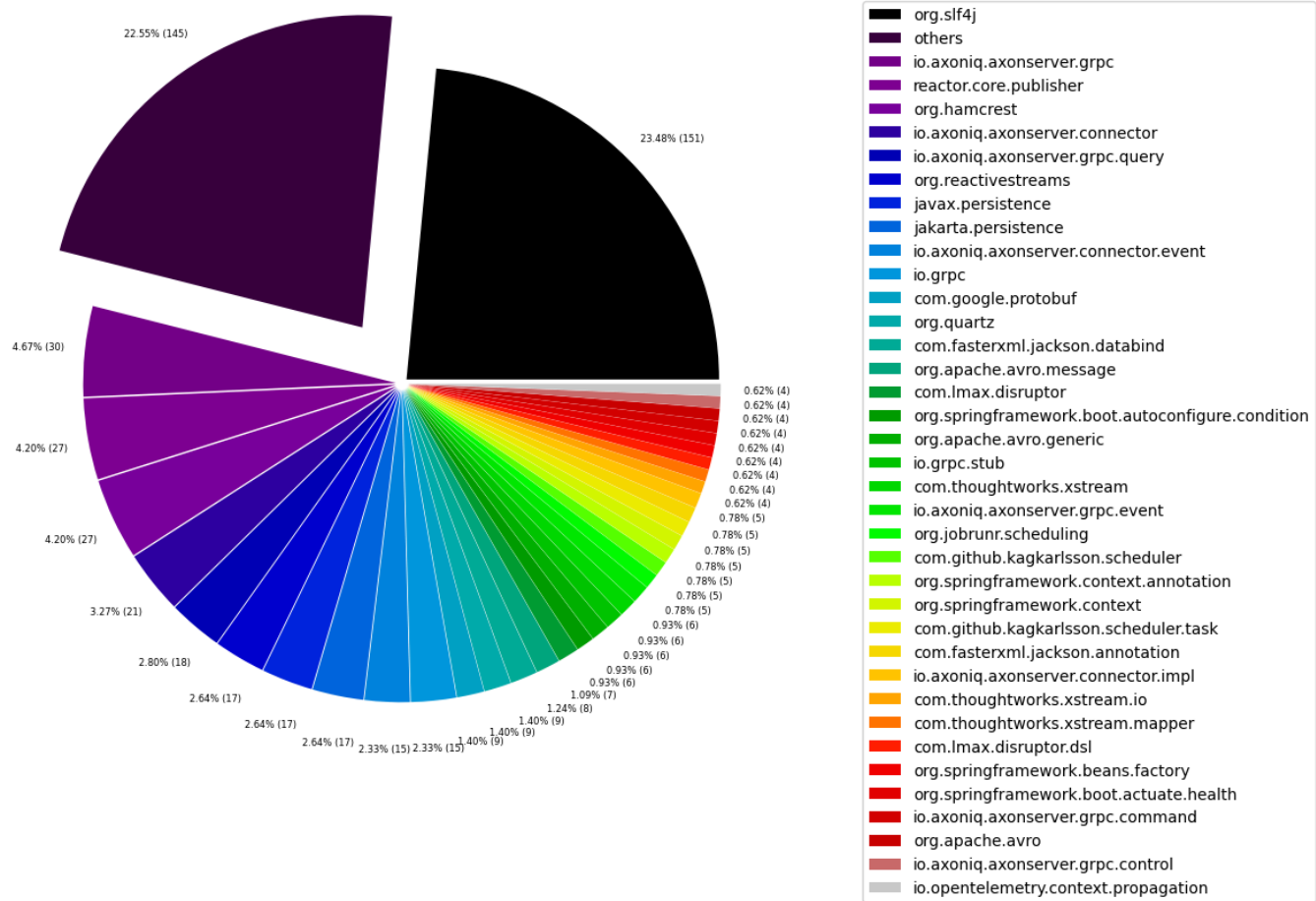
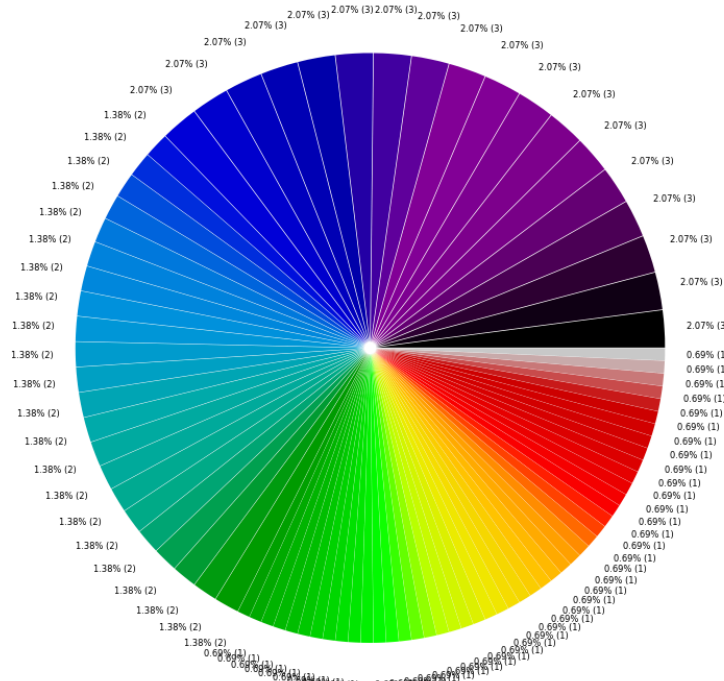


Table 3 Chart 1b - Most widely spread external packages in % by types (less than 0.5% overall "others" drill-down)

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)



io.opentelemetry.api.trace
com.thoughtworks.xstream.converters
org.dom4j
org.springframework.beans.factory.config
org.springframework.boot.autoconfigure
org.springframework.boot.autoconfigure.orm.jpa
org.springframework.boot.autoconfigure.service.connection
nu.xom
io.axoniq.axonserver.grpc.streams
io.axoniq.axonserver.connector.query
io.axoniq.axonserver.connector.control
com.thoughtworks.xstream.converters.collections
org.ehcache.event
org.testcontainers.containers
org.testcontainers.utility
com.fasterxml.jackson.core
com.fasterxml.jackson.databind.type
org.testcontainers.containers.wait.strategy
javax.cache.configuration
reactor.util.concurrent
org.junit.runners.model
org.axonframework.spring.config
org.springframework.beans
org.springframework.beans.factory.support
reactor.core.scheduler
com.fasterxml.jackson.databind.node
org.apache.avro.specific
net.sf.ehcache.event
javax.validation
net.sf.ehcache
org.springframework.core.type
org.springframework.core.env
com.thoughtworks.xstream.io.xml
com.github.kagkarlsson.scheduler.task.helper
com.fasterxml.jackson.dataformat.cbor.databind
jakarta.validation
io.grpc.netty.shaded.io.netty.util.internal
reactor.util.context
org.springframework.boot.testcontainers.service.connection
io.opentelemetry.context
org.springframework.boot.context.properties.bind
org.junit.runner
org.jobrunr.jobs.states
org.springframework.boot.docker.compose.core
org.junit.jupiter.api
org.junit.jupiter.api.extension
org.junit.rules
org.quartz.impl.matchers
org.springframework.util
org.springframework.core.annotation
org.springframework.transaction
org.ehcache.core
org.springframework.core.io
reactor.core
org.springframework.boot.docker.compose.service.connection
AggregateEventPublisherImpl
org.ehcache.config
javax.cache
com.codahale.metrics
com.fasterxml.jackson.databind.jsontype
com.fasterxml.jackson.databind.module
com.fasterxml.jackson.datatype.jsr310
com.google.common.base
com.google.gson
io.axoniq.axonserver.connector.admin
io.axoniq.axonserver.connector.command
io.grpc.netty.shaded.io.grpc.netty
io.grpc.netty.shaded.io.netty.handler.ssl
io.micrometer.core.instrument
io.micrometer.core.instrument.simple
io.opentelemetry.api
javax.cache.event
org.dom4j.io
org.apache.avro.io
org.apache.avro.util
org.apache.commons.lang3.tuple
org.axonframework.metrics
org.axonframework.micrometer
WeakValue
org.axonframework.spring.authorization
org.axonframework.spring.config.annotation
org.axonframework.spring.eventsourcing
org.axonframework.spring.jdbc
org.axonframework.spring.messaging.unitofwork
org.axonframework.spring.saga
org.axonframework.spring.serialization.support

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>

Top external package usage spread [%] by package (more than 0.5% overall)

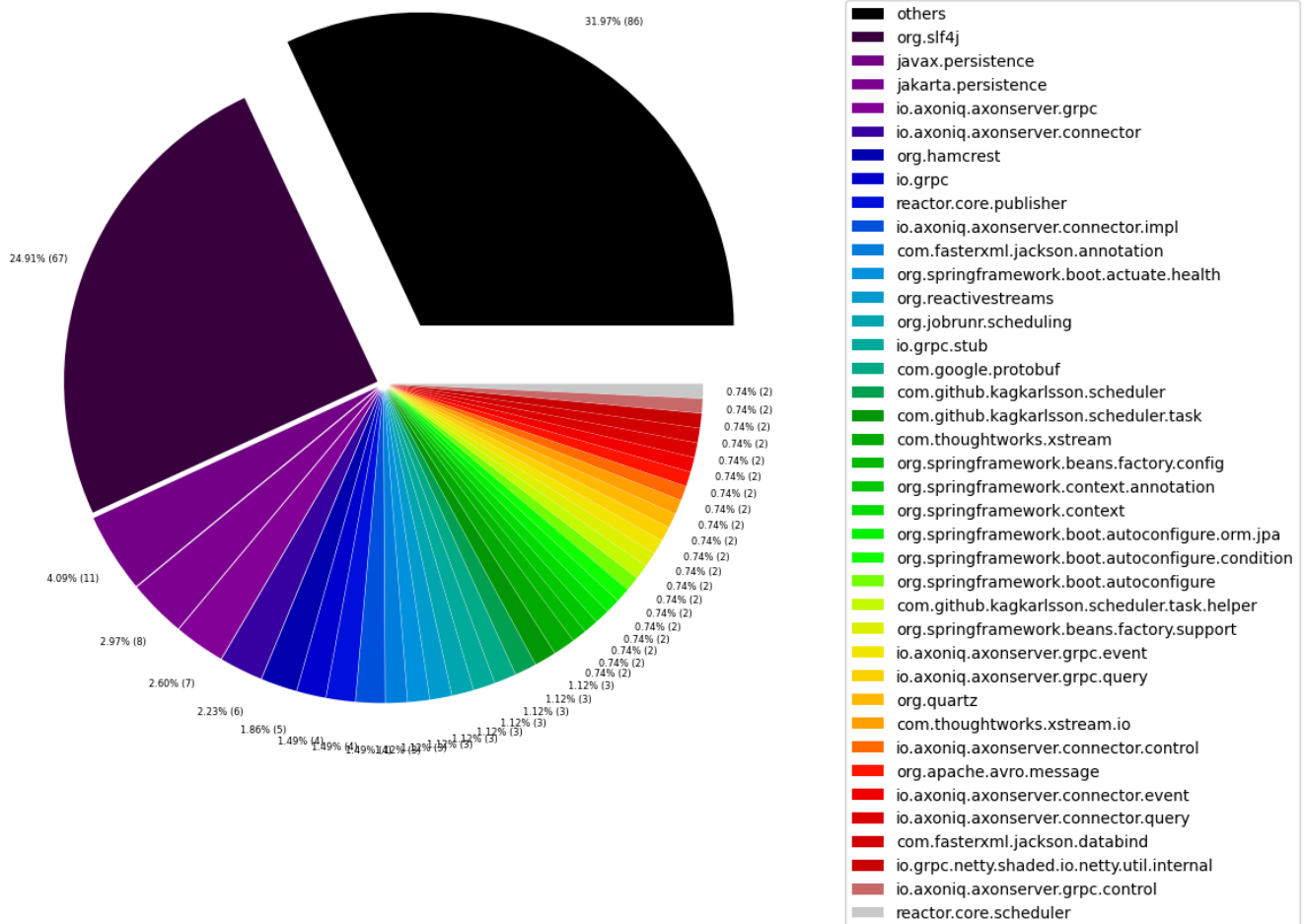


Table 3 Chart 2b - Most widely spread external packages in % by packages (less than 0.5% overall "others" drill-down)

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 0.7% overall "others" drill-down)

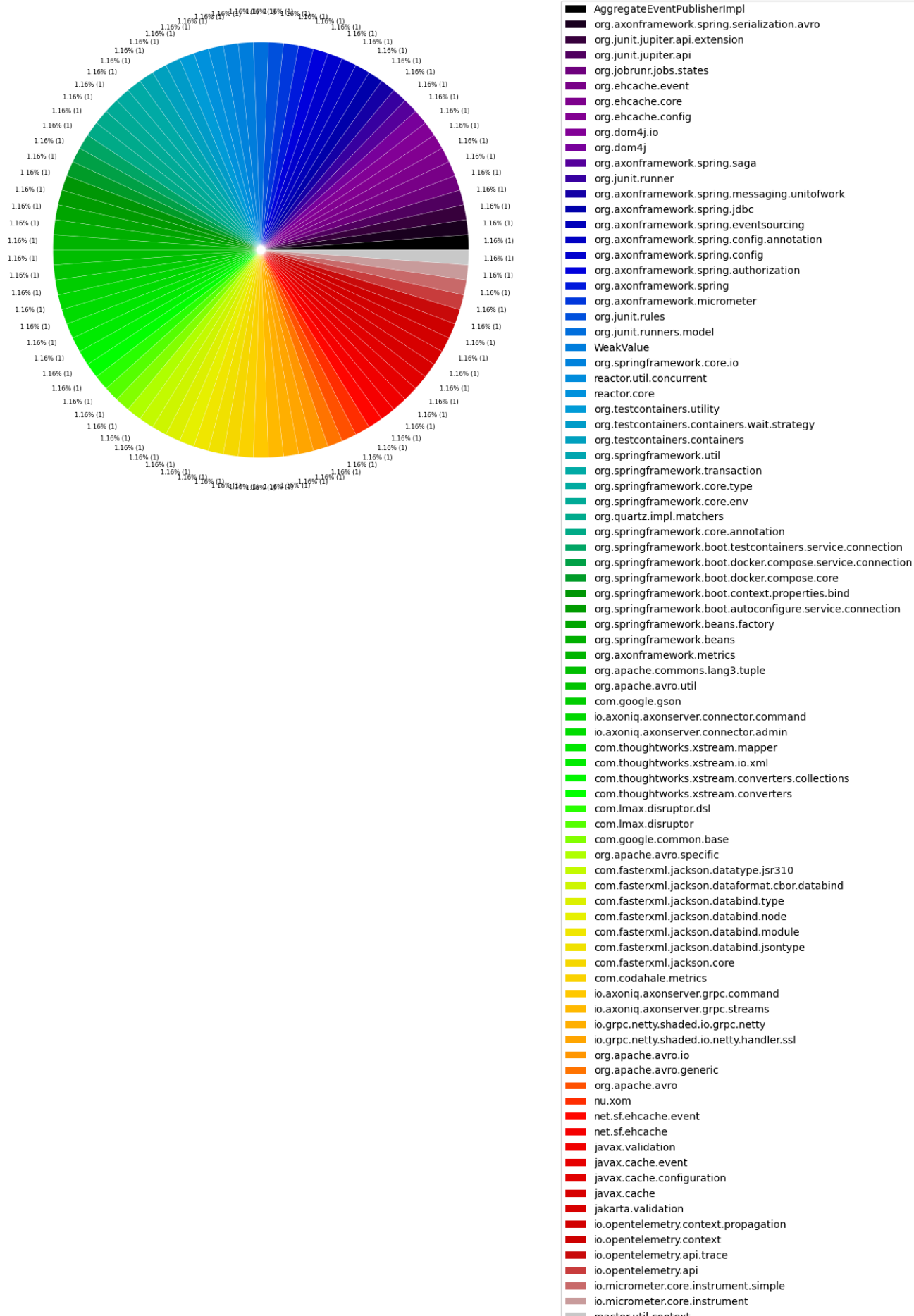


Table 4 - Top 20 most widely spread external packages grouped by their first 2 layers

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	medNumbe
0	org.slf4j	9	67	1	39	
1	jakarta.persistence	4	8	1	3	
2	javax.persistence	4	11	2	4	
3	com.fasterxml	2	5	1	4	
4	com.github	2	3	1	2	
5	com.google	2	4	1	3	
6	com.thoughtworks	2	3	1	2	
7	io.axoniq	2	9	1	8	
8	org.apache	2	2	1	1	
9	org.jobrunr	2	3	1	2	
10	org.reactivestreams	2	3	1	2	
11	reactor.core	2	5	2	3	
12	AggregateEventPublisherImpl	1	1	1	1	
13	WeakValue	1	1	1	1	
14	com.codahale	1	1	1	1	
15	com.lmax	1	1	1	1	
16	io.grpc	1	5	5	5	
17	io.micrometer	1	1	1	1	
18	io.opentelemetry	1	1	1	1	
19	jakarta.validation	1	1	1	1	

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

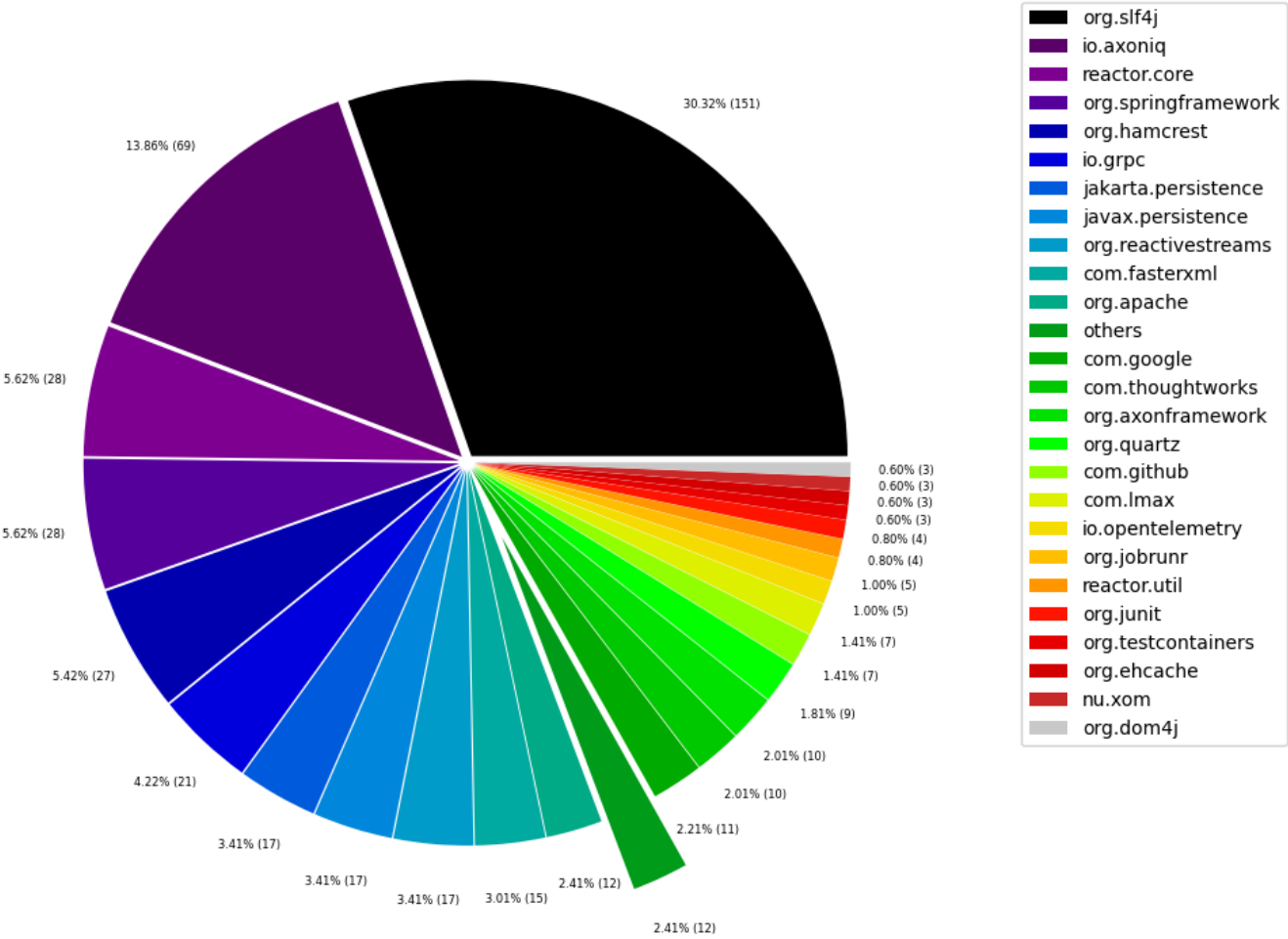


Table 4 Chart 1b - Most widely spread second level external packages in % by type (less than 0.5% overall "others" drill-down)

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)

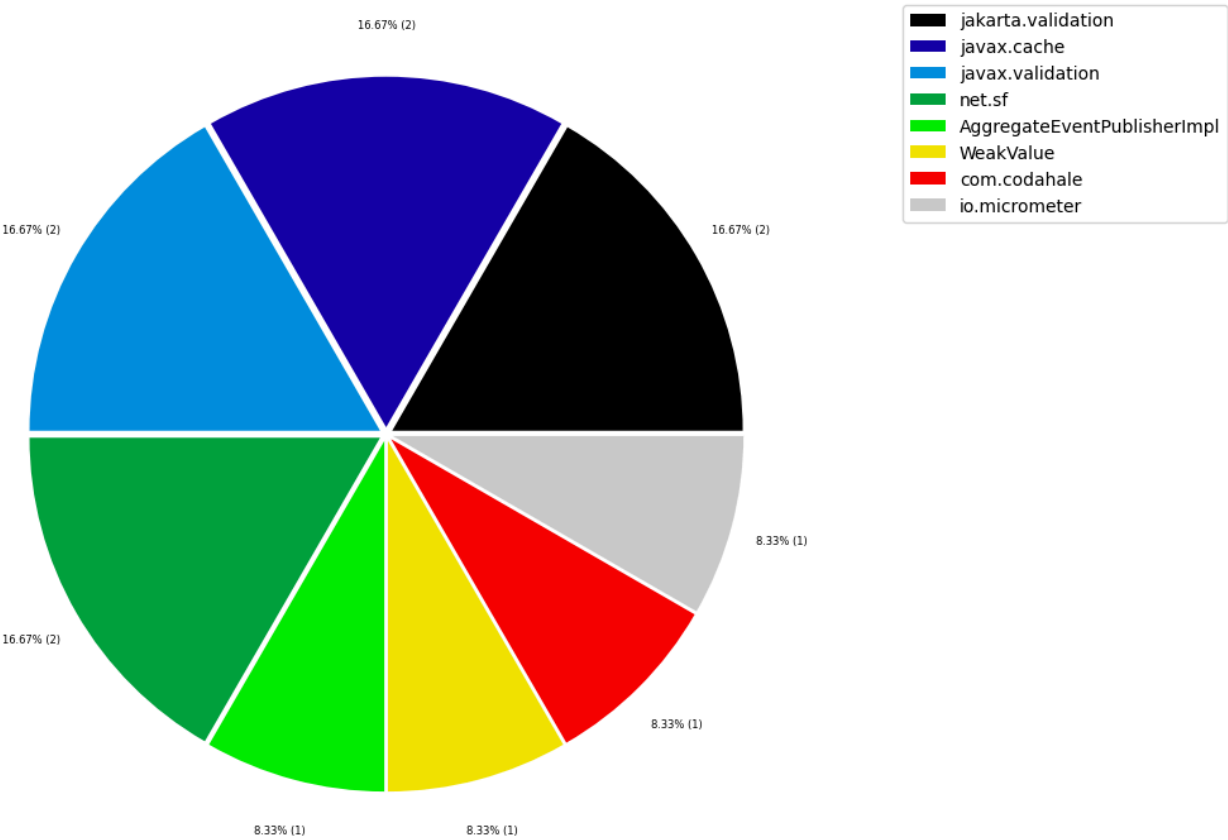


Table 4 Chart 2a - Most widely spread second level external packages in % by package (more than 0.5% overall)

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)

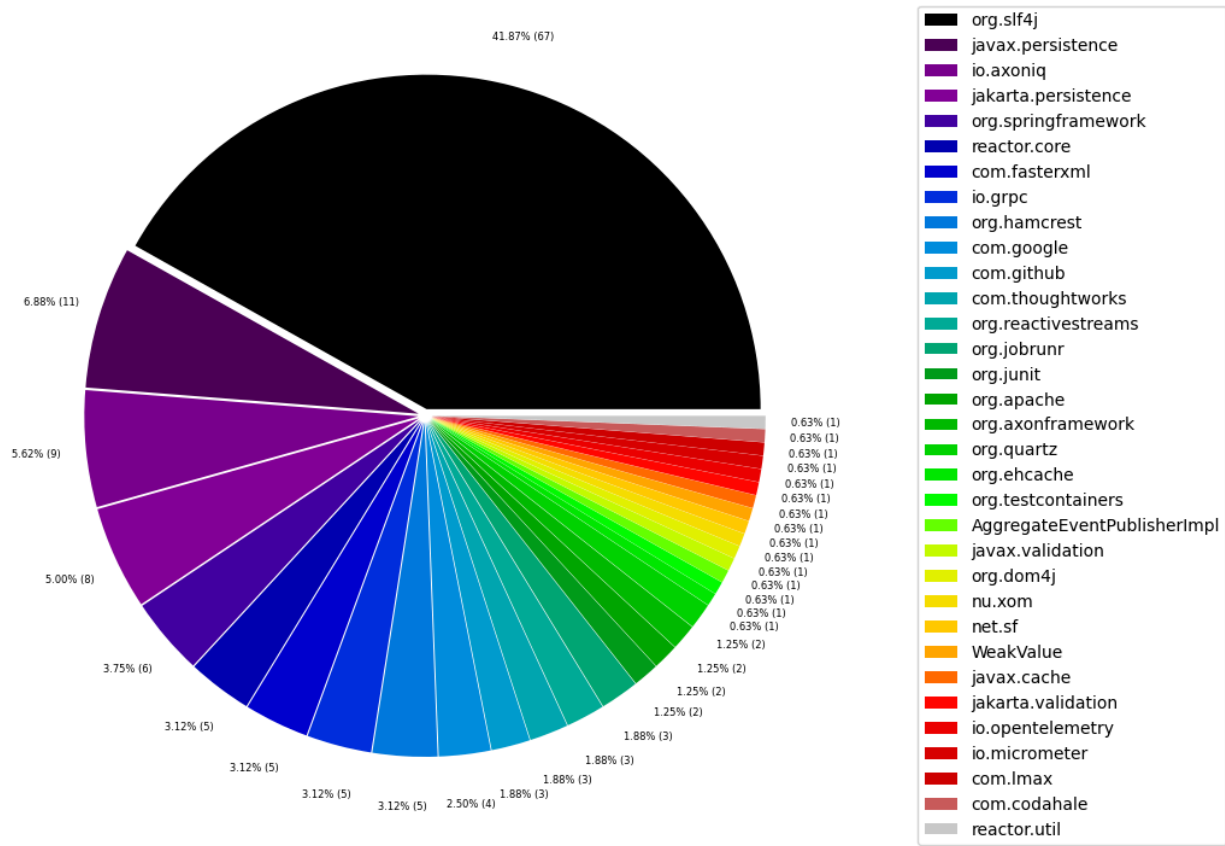


Table 4 Chart 2b - Most widely spread second level external packages in % by package (less than 0.5% overall "others" drill-down)

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

```
Received notification from DBMS server: {severity: WARNING} {code: Neo.ClientNotification.Statement.AggregationSkippedNull} {category: UNRECOGNIZED} {title: The query contains an aggregation function that skips null values.} {description: null value eliminated in set function.} {position: None} for query: '// External package usage per artifact sorted by external usage descending. Requires "Add_file_name and_extension.cyper".\n\n MATCH (artifact:Artifact:Archive)-[:CONTAINS]->(type:Type)\n OPTIONAL MATCH (type)-[:DEPENDS_ON]->(externalType:ExternalType)\n WITH artifact.name AS artifactName\n ,count(DISTINCT type.fqn) AS numberOfTypesInArtifact\n ,count(DISTINCT externalType.fqn) AS numberOfExternalTypesInArtifact\n ,count(DISTINCT replace(externalType.fqn, \'.\' + externalType.name, \'\'')) AS numberOfExternalPackagesInArtifact\n ,collect(DISTINCT type) AS typeList\n UNWIND typeList AS type\n MATCH (type)-[externalDependency:DEPENDS_ON]->(externalType:ExternalType)\n WITH numberOfTypesInArtifact\n ,numberOfExternalTypesInArtifact\n ,numberOfExternalPackagesInArtifact\n ,100.0 / numberOfTypesInArtifact * numberOfExternalTypesInArtifact AS externalTypeRate\n ,externalDependency\n ,artifactName\n ,type.fqn AS fullTypeName\n ,type.name AS typeName\n ,replace(externalType.fqn, \'.\' + externalType.name, \'\'') AS externalPackageName\n ,externalType.name AS externalTypeName\n WITH numberOfTypesInArtifact\n ,numberOfExternalTypesInArtifact\n ,numberOfExternalPackagesInArtifact\n ,externalTypeRate\n ,artifactName\n ,externalPackageName\n ,count(externalDependency) AS numberOfExternalTypeCaller\n ,sum(externalDependency.weight) AS numberOfExternalTypeCalls\n ,collect(DISTINCT externalTypeName) AS externalTypeNames\n RETURN artifactName\n ,externalPackageName\n ,numberOfExternalTypeCaller\n ,numberOfExternalTypeCalls\n ,numberOfTypesInArtifact\n ,numberOfExternalTypesInArtifact\n ,numberOfExternalPackagesInArtifact\n ,externalTypeRate\n ,externalTypeNames\n ORDER BY externalTypeRate DESC, artifactName ASC, numberOfExternalTypeCaller DESC, externalPackageName ASC'
```

	artifactName	externalPackageName	numberOfExternalTypeCaller	numberOfExternalTypeCalls	numberOfTypesInArti
0	axon-tracing-opentelemetry-4.11.0	io.opentelemetry.api.trace	9	47	
1	axon-tracing-opentelemetry-4.11.0	io.opentelemetry.context.propagation	9	18	
2	axon-tracing-opentelemetry-4.11.0	javax.annotation	3	8	
3	axon-tracing-opentelemetry-4.11.0	io.opentelemetry.context	2	7	
4	axon-tracing-opentelemetry-4.11.0	org.slf4j	2	7	
5	axon-tracing-opentelemetry-4.11.0	io.opentelemetry.api	1	2	
6	axon-spring-boot-autoconfigure-4.11.0	org.springframework.boot.autoconfigure.condition	75	168	
7	axon-spring-boot-autoconfigure-4.11.0	org.springframework.boot.autoconfigure	61	63	
8	axon-spring-boot-autoconfigure-4.11.0	org.springframework.context.annotation	48	152	
9	axon-spring-boot-autoconfigure-4.11.0	org.springframework.boot.context.properties	22	28	
10	axon-spring-boot-autoconfigure-4.11.0	org.springframework.beans.factory.annotation	11	28	
11	axon-spring-boot-autoconfigure-4.11.0	javax.annotation	8	9	
12	axon-spring-boot-autoconfigure-4.11.0	org.slf4j	8	13	
13	axon-spring-boot-autoconfigure-4.11.0	org.springframework.boot.actuate.health	7	24	
14	axon-spring-boot-autoconfigure-4.11.0	org.axonframework.spring.config	6	18	
15	axon-spring-boot-autoconfigure-4.11.0	org.springframework.beans.factory	6	21	
16	axon-spring-boot-autoconfigure-4.11.0	org.springframework.context	6	24	
17	axon-spring-boot-autoconfigure-4.11.0	jakarta.persistence	4	6	
18	axon-spring-boot-autoconfigure-4.11.0	javax.persistence	4	6	
19	axon-spring-boot-autoconfigure-4.11.0	org.apache.avro.message	4	10	
20	axon-spring-boot-autoconfigure-4.11.0	org.springframework.beans.factory.config	4	7	
21	axon-spring-boot-autoconfigure-4.11.0	org.springframework.beans.factory.support	4	21	
22	axon-spring-boot-	org.springframework.boot.testcontainers.servic...	4	9	

	artifactName	externalPackageName	numberOfExternalTypeCaller	numberOfExternalTypeCalls	numberOfTypesInArti
	autoconfigure-4.11.0				
23	axon-spring-boot-autoconfigure-4.11.0	org.springframework.core.annotation	4	4	
24	axon-spring-boot-autoconfigure-4.11.0	org.axonframework.spring.serialization.avro	3	8	
25	axon-spring-boot-autoconfigure-4.11.0	org.springframework.boot.autoconfigure.orm.jpa	3	3	
26	axon-spring-boot-autoconfigure-4.11.0	org.springframework.boot.autoconfigure.service...	3	3	
27	axon-spring-boot-autoconfigure-4.11.0	org.springframework.boot.context.properties.bind	3	4	
28	axon-spring-boot-autoconfigure-4.11.0	com.fasterxml.jackson.databind	2	10	
29	axon-spring-boot-autoconfigure-4.11.0	com.fasterxml.jackson.dataformat.cbor.databind	2	8	
30	axon-spring-boot-autoconfigure-4.11.0	com.thoughtworks.xstream	2	8	
31	axon-spring-boot-autoconfigure-4.11.0	org.axonframework.metrics	2	8	
32	axon-spring-boot-autoconfigure-4.11.0	org.axonframework.micrometer	2	10	
33	axon-spring-boot-autoconfigure-4.11.0	org.axonframework.spring.config.annotation	2	7	
34	axon-spring-boot-autoconfigure-4.11.0	org.axonframework.spring.eventsourcing	2	10	
35	axon-spring-boot-autoconfigure-4.11.0	org.springframework.beans	2	3	
36	axon-spring-boot-autoconfigure-4.11.0	org.springframework.boot.docker.compose.core	2	5	
37	axon-spring-boot-autoconfigure-4.11.0	org.springframework.boot.docker.compose.servic...	2	5	
38	axon-spring-boot-autoconfigure-4.11.0	org.springframework.core.env	2	2	
39	axon-spring-boot-autoconfigure-4.11.0	org.springframework.core.type	2	4	

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

```

Received notification from DBMS server: {severity: WARNING} {code: Neo.ClientNotification.Statement.AggregationSkippedNull} {category: UNRECOGNIZED} {title: The query contains an aggregation function that skips null values.} {description: null value eliminated in set function.} {position: None} for query: '// External package usage per artifact and external package. Requires "Add_file_name and_extension.cypher".\n\n// Get the overall artifact statistics first\n MATCH (artifact:Artifact)-[:CONTAINS]->(package:Package)\n MATCH (package)-[:CONTAINS]->(type:Type)\n OPTIONAL MATCH (packageUsingExternal:Package)-[:CONTAINS]->(type)-[:DEPENDS_ON]->(external:ExternalType)\n WITH artifact.name AS artifactName\n ,count(DISTINCT package.fqn) AS artifactPackages\n ,count(DISTINCT type.fqn) AS artifactTypes\n ,count(DISTINCT replace(external.fqn, '\\.' + external.name, '\\\\')) AS artifactExternalPackages\n ,count(DISTINCT packageUsingExternal.fqn) AS artifactExternalCallingPackages\n ,collect(type) AS typeList\n WITH artifactName\n ,artifactPackages\n ,artifactTypes\n ,artifactExternalPackages\n ,artifactExternalCallingPackages\n ,round((100.0 / artifactPackages * artifactExternalCallingPackages), 2) AS artifactExternalCallingPackagesRate\n ,typeList\n // Get the external dependencies for each internal type\n UNWIND typeList AS type\n MATCH (type)-[:DEPENDS_ON]->(externalType:ExternalType)\n MATCH (typePackage:Package)-[:CONTAINS]->(type)\n // Optionally filter out dependencies to external annotations\n // WHERE NOT externalType:ExternalAnnotation\n WITH artifactName\n ,artifactPackages\n ,artifactTypes\n ,artifactExternalPackages\n ,artifactExternalCallingPackages\n ,artifactExternalCallingPackagesRate\n ,typePackage.fqn AS packageName\n ,type.fqn AS fullTypeName\n ,replace(externalType.fqn, '\\.' + externalType.name, '\\\\') AS externalPackageName\n // Group by artifact and external package\n RETURN artifactName\n ,artifactPackages\n ,artifactTypes\n ,artifactExternalPackages\n ,artifactExternalCallingPackages\n ,artifactExternalCallingPackagesRate\n ,externalPackageName\n ,count(DISTINCT packageName) AS numberOfPackages\n ,count(DISTINCT fullTypeName) AS numberOfTypes\n ,100.0 / artifactPackages * count(DISTINCT packageName) AS packagesCallingExternalRate\n ,100.0 / artifactTypes * count(DISTINCT fullTypeName) AS typesCallingExternalRate\n ,COLLECT(DISTINCT packageName) AS nameOfPackages\n ,COLLECT(DISTINCT fullTypeName)[0..9] AS someTypeNames\n // Order the results by number of packages that use the external package dependency descending\n ORDER BY artifactExternalCallingPackagesRate DESC, artifactName ASC, numberOfPackages DESC, externalPackageName ASC'

```

	artifactName	artifactPackages	artifactTypes	artifactExternalPackages	artifactExternalCallingPackages	artifactExternalCallingPackagesRat
0	axon-configuration-4.11.0	1	41	2	1	100.
1	axon-configuration-4.11.0	1	41	2	1	100.
2	axon-disruptor-4.11.0	1	22	5	1	100.
3	axon-disruptor-4.11.0	1	22	5	1	100.
4	axon-disruptor-4.11.0	1	22	5	1	100.
5	axon-disruptor-4.11.0	1	22	5	1	100.
6	axon-disruptor-4.11.0	1	22	5	1	100.
7	axon-spring-boot-autoconfigure-4.11.0	9	88	52	9	100.
8	axon-spring-boot-autoconfigure-4.11.0	9	88	52	9	100.
9	axon-spring-boot-autoconfigure-4.11.0	9	88	52	9	100.
10	axon-spring-boot-autoconfigure-4.11.0	9	88	52	9	100.
11	axon-spring-boot-autoconfigure-4.11.0	9	88	52	9	100.
12	axon-spring-boot-autoconfigure-4.11.0	9	88	52	9	100.
13	axon-spring-boot-autoconfigure-4.11.0	9	88	52	9	100.
14	axon-spring-boot-autoconfigure-4.11.0	9	88	52	9	100.
15	axon-spring-boot-autoconfigure-4.11.0	9	88	52	9	100.
16	axon-spring-boot-autoconfigure-4.11.0	9	88	52	9	100.
17	axon-spring-boot-autoconfigure-4.11.0	9	88	52	9	100.
18	axon-spring-boot-autoconfigure-4.11.0	9	88	52	9	100.
19	axon-spring-boot-autoconfigure-4.11.0	9	88	52	9	100.
20	axon-spring-boot-autoconfigure-4.11.0	9	88	52	9	100.
21	axon-spring-boot-autoconfigure-4.11.0	9	88	52	9	100.
22	axon-spring-boot-	9	88	52	9	100.

	artifactName	artifactPackages	artifactTypes	artifactExternalPackages	artifactExternalCallingPackages	artifactExternalCallingPackagesRat
	autoconfigure-4.11.0					
23	axon-spring-boot-autoconfigure-4.11.0	9	88	52	9	100.
24	axon-spring-boot-autoconfigure-4.11.0	9	88	52	9	100.
25	axon-spring-boot-autoconfigure-4.11.0	9	88	52	9	100.
26	axon-spring-boot-autoconfigure-4.11.0	9	88	52	9	100.
27	axon-spring-boot-autoconfigure-4.11.0	9	88	52	9	100.
28	axon-spring-boot-autoconfigure-4.11.0	9	88	52	9	100.
29	axon-spring-boot-autoconfigure-4.11.0	9	88	52	9	100.

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`.


```

Received notification from DBMS server: {severity: WARNING} {code: Neo.ClientNotification.Statement.AggregationSkippedNull} {category: UNRECOGNIZED} {title: The query contains an aggregation function that skips null values.} {description: null value eliminated in set function.} {position: None} for query: '// External second level package usage per artifact and external package. Requires "Add_file_name and_extension.cyphe
r".\n\n// Get the overall artifact statistics first\n  MATCH (artifact:Artifact)-[:CONTAINS]->(package:Package)\n  MATCH (package)-[:CONTAINS]->(type:Type)\n  OPTIONAL MATCH (packageUsingExternal:Package)-[:CONTAINS]->(type)-[:DEPENDS_ON]->(external:ExternalType)\n    WITH artifact.name AS artifactName\n    ,count(DISTINCT package.fqn) AS artifactPackages\n    ,count(DISTINCT type.fqn) AS artifactTypes\n    ,count(DISTINCT split(external.fqn,\'.\') [0..2]) AS artifactExternalPackagesFirst2Levels\n    ,count(DISTINCT packageUsingExternal.fqn) AS artifactExternalCallingPackages\n    ,collect(type) AS typeList\n  WITH artifactName\n    ,artifactPackages\n    ,artifactTypes\n    ,artifactExternalPackagesFirst2Levels\n    ,artifactExternalCallingPackages\n    ,round((100.0 / artifactPackages * artifactExternalCallingPackages), 2) AS artifactExternalCallingPackagesRate\n    ,typeList\n  // Get the external dependencies for each internal type\n  UNWIND typeList AS type\n  MATCH (type)-[:DEPENDS_ON]->(externalType:ExternalType)\n  MATCH (typePackage:Package)-[:CONTAINS]->(type)\n  // Optionally filter out dependencies to external annotations\n  // WHERE NOT externalType:ExternalAnnotation\n  WITH artifactName\n    ,artifactPackages\n    ,artifactTypes\n    ,artifactExternalPackagesFirst2Levels\n    ,artifactExternalCallingPackages\n    ,artifactExternalCallingPackagesRate\n    ,typePackage.fqn AS packageName\n    ,type.fqn AS fullTypeName\n    ,apoc.text.join(split(externalType.fqn,\'.\') [0..2], \'..\') AS externalPackageNameFirst2Levels\n  // Group by artifact and first to external package levels\n  RETURN artifactName\n    ,artifactPackages\n    ,artifactTypes\n    ,artifactExternalPackagesFirst2Levels\n    ,artifactExternalCallingPackages\n    ,artifactExternalCallingPackagesRate\n    ,externalPackageNameFirst2Levels\n    ,count(DISTINCT packageName) AS numberOfPackages\n    ,count(DISTINCT fullTypeName) AS numberOfTypes\n    ,100.0 / artifactPackages * count(DISTINCT packageName) AS packagesCallingExternalRate\n    ,100.0 / artifactTypes * count(DISTINCT fullTypeName) AS typesCallingExternalRate\n    ,COLLECT(DISTINCT packageName) AS nameOfPackages\n    ,COLLECT(DISTINCT fullTypeName)[0..9] AS someTypeNames\n  // Order the results by number of packages that use the external package dependency descending\n  ORDER BY artifactExternalCallingPackagesRate DESC, artifactName ASC, numberOfPackages DESC, externalPackageNameFirst2Levels ASC'

```

	artifactName	artifactPackages	artifactTypes	artifactExternalPackagesFirst2Levels	artifactExternalCallingPackages	artifactExternalCalling
0	axon-configuration-4.11.0	1	41	2	1	
1	axon-configuration-4.11.0	1	41	2	1	
2	axon-disruptor-4.11.0	1	22	4	1	
3	axon-disruptor-4.11.0	1	22	4	1	
4	axon-disruptor-4.11.0	1	22	4	1	
5	axon-disruptor-4.11.0	1	22	4	1	
6	axon-spring-boot-autoconfigure-4.11.0	9	88	15	9	
7	axon-spring-boot-autoconfigure-4.11.0	9	88	15	9	
8	axon-spring-boot-autoconfigure-4.11.0	9	88	15	9	
9	axon-spring-boot-autoconfigure-4.11.0	9	88	15	9	
10	axon-spring-boot-autoconfigure-4.11.0	9	88	15	9	
11	axon-spring-boot-autoconfigure-4.11.0	9	88	15	9	
12	axon-spring-boot-autoconfigure-4.11.0	9	88	15	9	
13	axon-spring-boot-autoconfigure-4.11.0	9	88	15	9	
14	axon-spring-boot-autoconfigure-4.11.0	9	88	15	9	
15	axon-spring-boot-autoconfigure-4.11.0	9	88	15	9	
16	axon-spring-boot-autoconfigure-4.11.0	9	88	15	9	
17	axon-spring-boot-autoconfigure-4.11.0	9	88	15	9	
18	axon-spring-boot-autoconfigure-4.11.0	9	88	15	9	
19	axon-spring-boot-autoconfigure-4.11.0	9	88	15	9	
20	axon-spring-boot-autoconfigure-4.11.0	9	88	15	9	
21	axon-tracing-opentelemetry-4.11.0	1	5	3	1	
22	axon-tracing-opentelemetry-4.11.0	1	5	3	1	
23	axon-tracing-opentelemetry-4.11.0	1	5	3	1	

	artifactName	artifactPackages	artifactTypes	artifactExternalPackagesFirst2Levels	artifactExternalCallingPackages	artifactExternalCalling
24	axon-server-connector-4.11.0	11	142	8	10	
25	axon-server-connector-4.11.0	11	142	8	10	
26	axon-server-connector-4.11.0	11	142	8	10	
27	axon-server-connector-4.11.0	11	142	8	10	
28	axon-server-connector-4.11.0	11	142	8	10	
29	axon-server-connector-4.11.0	11	142	8	10	

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.

	axon-messaging-4.11.0	axon-spring-boot-autoconfigure-4.11.0	axon-server-connector-4.11.0	axon-test-4.11.0	axon-modelling-4.11.0	eventsourcing-4.11.0	axon-tracing-opentelemetry-4.11.0	axon-disruptor-4.11.0	conf
externalPackageName									
AggregateEventPublisherImpl	0	0	0	1	0	0	0	0	
WeakValue	0	0	0	0	0	0	0	1	
com.codahale.metrics	0	1	0	0	0	0	0	0	
com.fasterxml.jackson.annotation	11	0	0	0	2	0	0	0	
com.fasterxml.jackson.core	1	0	0	0	0	0	0	0	
...	
reactor.core	1	0	0	0	0	0	0	0	
reactor.core.publisher	2	0	2	0	0	0	0	0	
reactor.core.scheduler	0	0	2	0	0	0	0	0	
reactor.util.concurrent	1	0	0	0	0	0	0	0	
reactor.util.context	1	0	0	0	0	0	0	0	

129 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-4.11.0	axon-server-connector-4.11.0	axon-spring-boot-autoconfigure-4.11.0	axon-modelling-4.11.0	event-sourcing-4.11.0	axon-test-4.11.0	axon-disruptor-4.11.0	axon-tracing-opentelemetry-4.11.0	con
externalPackageNameFirst2Levels									
AggregateEventPublisherImpl	0	0	0	0	0	1	0	0	
WeakValue	0	0	0	0	0	0	1	0	
com.codahale	0	0	1	0	0	0	0	0	
com.fasterxml	12	0	1	2	0	0	0	0	
com.github	2	0	1	0	0	0	0	0	
com.google	0	3	0	0	0	1	0	0	
com.lmax	0	0	0	0	0	0	1	0	
com.thoughtworks	2	0	1	0	0	0	0	0	
io.axoniq	0	8	1	0	0	0	0	0	
io.grpc	0	5	0	0	0	0	0	0	
io.micrometer	0	0	1	0	0	0	0	0	
io.opentelemetry	0	0	0	0	0	0	0	1	
jakarta.persistence	4	0	2	2	2	0	0	0	
jakarta.validation	1	0	0	0	0	0	0	0	
javax.annotation	49	8	4	4	6	4	1	1	
javax.cache	1	0	0	0	0	0	0	0	
javax.persistence	7	0	2	3	3	0	0	0	
javax.validation	1	0	0	0	0	0	0	0	
net.sf	1	0	0	0	0	0	0	0	
nu.xom	1	0	0	0	0	0	0	0	
org.apache	1	0	1	0	0	0	0	0	
org.axonframework	0	0	2	0	0	0	0	0	
org.dom4j	1	0	0	0	0	0	0	0	
org.ehcache	1	0	0	0	0	0	0	0	
org.hamcrest	0	0	0	0	0	5	0	0	
org.jetbrains	0	0	1	0	0	0	0	0	
org.jobrunr	2	0	1	0	0	0	0	0	
org.junit	0	0	0	0	0	2	0	0	
org.quartz	2	0	0	0	0	0	0	0	
org.reactivestreams	2	1	0	0	0	0	0	0	
org.slf4j	39	9	3	6	5	2	1	1	
org.springframework	0	1	7	0	0	0	0	0	
org.testcontainers	0	0	0	0	0	1	0	0	
reactor.core	2	3	0	0	0	0	0	0	
reactor.util	1	0	0	0	0	0	0	0	

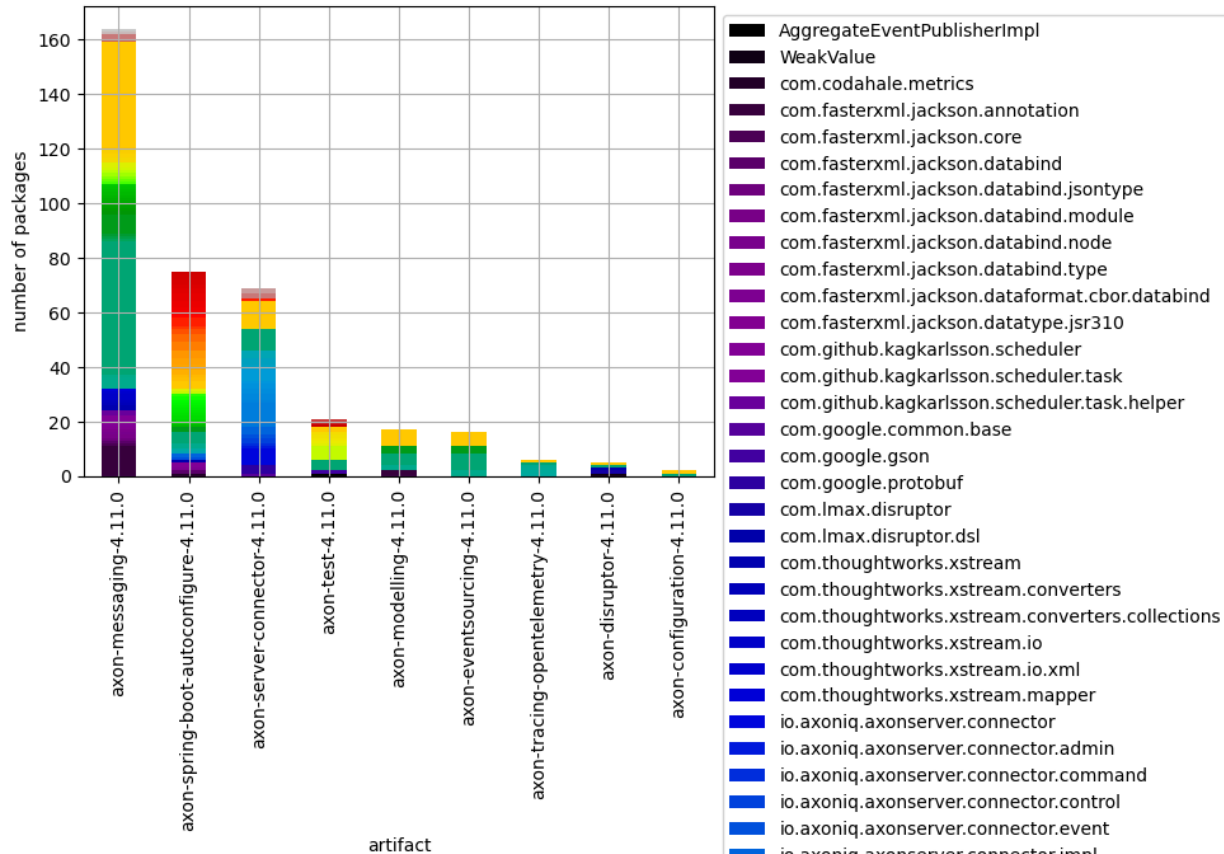
Table 7 Chart 1 - Top 15 external dependency using artifacts and their external packages stacked

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.apache.avro
- org.apache.avro.generic
- org.apache.avro.io
- org.apache.avro.message
- org.apache.avro.specific
- org.apache.avro.util
- org.apache.commons.lang3.tuple
- org.axonframework.metrics
- org.axonframework.micrometer

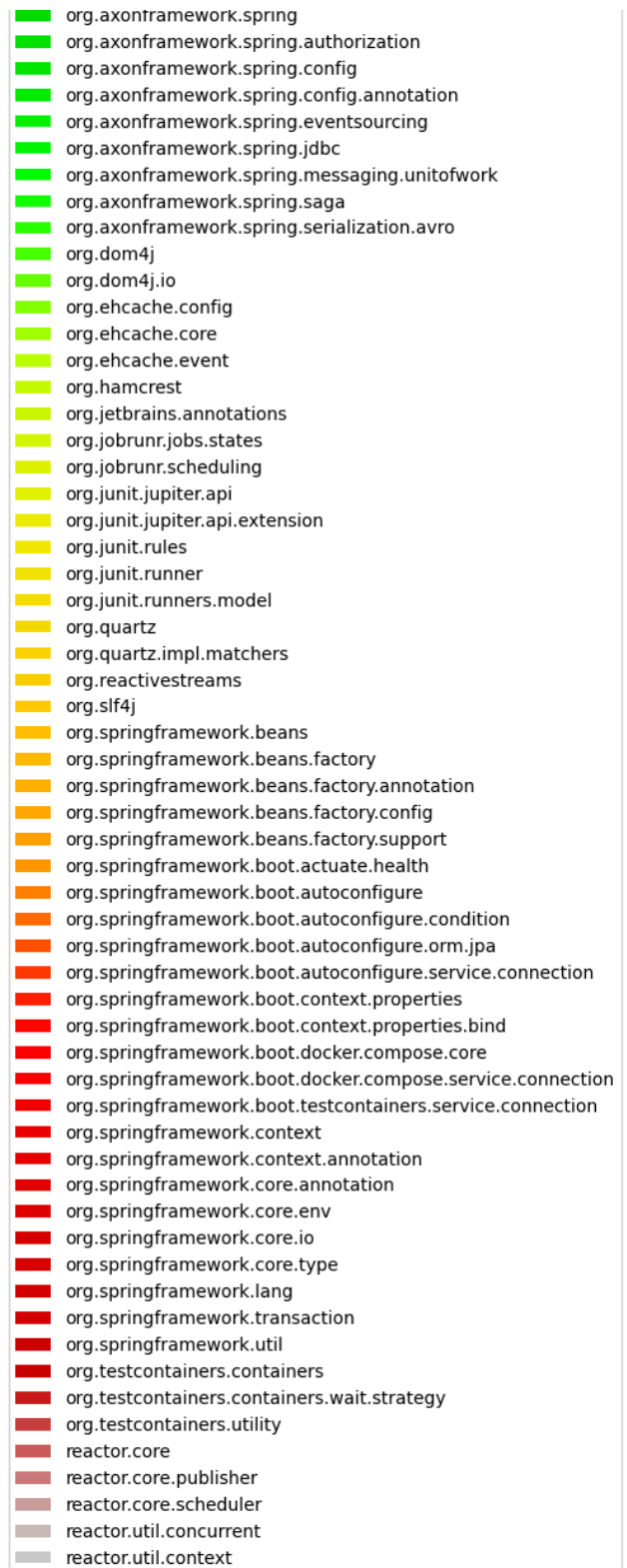


Table 7 Chart 2 - Top 15 external dependency using artifacts and their external packages (first 2 levels) stacked

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>

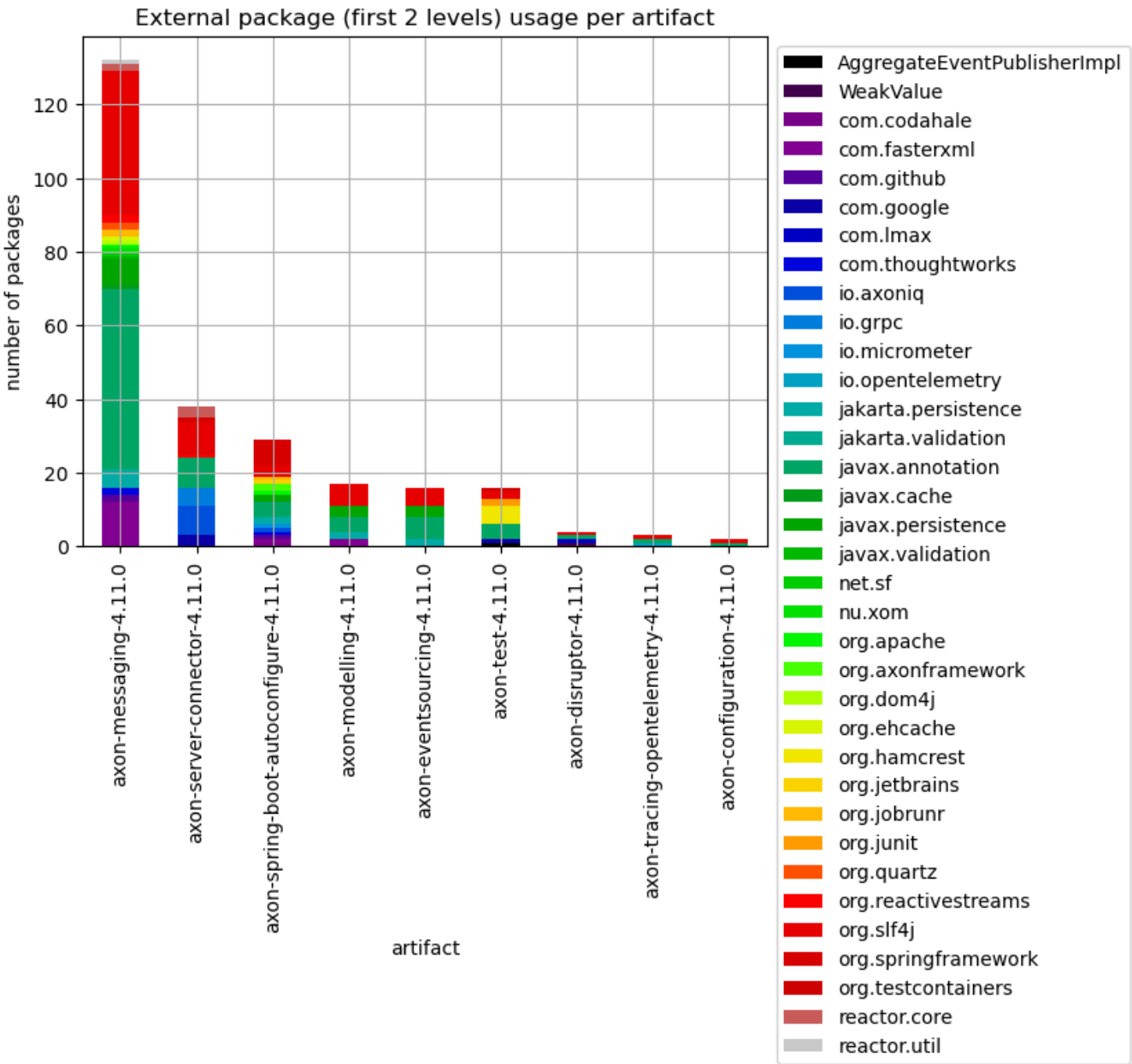


Table 8 - External usage per artifact

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


```

Received notification from DBMS server: {severity: WARNING} {code: Neo.ClientNotification.Statement.AggregationSkippedNull} {category: UNRECOGNIZED} {title: The query contains an aggregation function that skips null values.} {description: null value eliminated in set function.} {position: None} for query: '// External package usage per artifact top externals. Requires "Add_file_name and_extension.cypher".\n\n MATCH (artifact:Artifact:Archive)-[:CONTAINS]->(type:Type)\n OPTIONAL MATCH (type)-[:DEPENDS_ON]->(externalType:ExternalType)\n WITH artifact.name AS artifactName\n ,count(DISTINCT type.fqn) AS numberOfTypesInArtifact\n ,count(DISTINCT externalType.fqn) AS numberOfExternalTypesInArtifact\n ,count(DISTINCT replace(externalType.fqn, \'.\' + externalType.name, \'\'')) AS numberOfExternalPackagesInArtifact\n ,collect(DISTINCT type) AS typeList\n UNWIND typeList AS type\n MATCH (type)-[externalDependency:DEPENDS_ON]->(externalType:ExternalType)\n WITH numberOfTypesInArtifact\n ,numberOfExternalTypesInArtifact\n ,numberOfExternalPackagesInArtifact\n ,100.0 / numberOfTypesInArtifact * numberOfExternalTypesInArtifact AS externalTypeRate\n ,externalDependency\n ,artifactName\n ,type.fqn\n AS fullTypeName\n ,type.name AS typeName\n ,replace(externalType.fqn, \'.\' + externalType.name, \'\'') AS externalPackageName\n ,externalType.name AS externalTypeName\n ORDER BY externalTypeRate DESC, artifactName ASC\n WITH numberOfTypesInArtifact\n ,numberOfExternalTypesInArtifact\n ,numberOfExternalPackagesInArtifact\n ,externalTypeRate\n ,artifactName\n ,externalPackageName\n ,count(externalDependency) AS numberOfExternalTypeCaller\n ,sum(externalDependency.weight) AS numberOfExternalTypeCalls\n ,collect(DISTINCT externalTypeName) AS externalTypeNames\n ORDER BY externalTypeRate DESC, artifactName ASC, numberOfExternalTypeCaller DESC\n WITH numberOfTypesInArtifact\n ,numberOfExternalTypesInArtifact\n ,numberOfExternalPackagesInArtifact\n ,externalTypeRate\n ,artifactName\n ,COLLECT(DISTINCT externalPackageName) AS externalPackageNames\n ,SUM(numberOfExternalTypeCaller) AS numberOfExternalTypeCaller\n ,sum(numberOfExternalTypeCalls) AS numberOfExternalTypeCalls\n ,collect(externalTypeNames) AS externalTypeNames\n\n RETURN artifactName\n ,numberOfTypesInArtifact\n ,numberOfExternalTypesInArtifact\n ,numberOfExternalPackagesInArtifact\n ,externalTypeRate\n ,numberOfExternalTypeCaller\n ,numberOfExternalTypeCalls\n ,size(externalPackageNames) AS numberOfExternalPackages\n ,externalPackageNames[0..4] AS top5ExternalPackages\n ,apoc.coll.flatten(externalTypeNames)[0..9] AS someExternalTypes'

```

	artifactName	numberOfTypesInArtifact	numberOfExternalTypesInArtifact	numberOfExternalPackagesInArtifact	externalTypeRate	numberOfExternalTypeCaller	numberOfExternalTypeCalls	externalTypeNames
0	axon-tracing-opentelemetry-4.11.0	5	16	6	320.000000			
1	axon-spring-boot-autoconfigure-4.11.0	88	112	52	127.272727			
2	axon-server-connector-4.11.0	142	115	26	80.985915			
3	axon-disruptor-4.11.0	22	13	5	59.090909			
4	axon-test-4.11.0	87	28	13	32.183908			
5	axon-modelling-4.11.0	158	35	5	22.151899			
6	axon-messaging-4.11.0	809	179	50	22.126082			
7	axon-eventsourcing-4.11.0	133	25	4	18.796992			
8	axon-configuration-4.11.0	41	4	2	9.756098			

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	numberOfExte
0	axon-server-connector-4.11.0	org.axonframework.axonserver.connector.util	io.grpc	50	
1	axon-test-4.11.0	org.axonframework.test.matchers	org.hamcrest	38	
2	axon-messaging-4.11.0	org.axonframework.queryhandling	reactor.core.publisher	28	
3	axon-server-connector-4.11.0	org.axonframework.axonserver.connector.query	io.axoniq.axonserver.grpc.query	25	
4	axon-server-connector-4.11.0	org.axonframework.axonserver.connector.event.axon	io.axoniq.axonserver.connector.event	21	
5	axon-server-connector-4.11.0	org.axonframework.axonserver.connector.query.s...	io.axoniq.axonserver.grpc.query	21	
6	axon-server-connector-4.11.0	org.axonframework.axonserver.connector.util	io.axoniq.axonserver.grpc	20	
7	axon-messaging-4.11.0	org.axonframework.deadline.quartz	org.quartz	18	
8	axon-messaging-4.11.0	org.axonframework.eventhandling.scheduling.quartz	org.quartz	17	
9	axon-server-connector-4.11.0	org.axonframework.axonserver.connector.query	io.axoniq.axonserver.connector	17	
10	axon-messaging-4.11.0	org.axonframework.eventhandling	org.slf4j	16	
11	axon-messaging-4.11.0	org.axonframework.serialization.avro	org.apache.avro	16	
12	axon-messaging-4.11.0	org.axonframework.eventhandling.pooled	org.slf4j	15	
13	axon-messaging-4.11.0	org.axonframework.serialization.json	com.fasterxml.jackson.databind	15	
14	axon-disruptor-4.11.0	org.axonframework.disruptor.commandhandling	org.slf4j	12	
15	axon-server-connector-4.11.0	org.axonframework.axonserver.connector.event.axon	org.slf4j	12	
16	axon-configuration-4.11.0	org.axonframework.config	org.slf4j	11	
17	axon-disruptor-4.11.0	org.axonframework.disruptor.commandhandling	com.lmax.disruptor	9	
18	axon-messaging-4.11.0	org.axonframework.queryhandling	org.slf4j	9	
19	axon-server-connector-4.11.0	org.axonframework.axonserver.connector.command	io.axoniq.axonserver.grpc.command	9	
20	axon-server-connector-4.11.0	org.axonframework.axonserver.connector.query	io.axoniq.axonserver.grpc	9	
21	axon-tracing-opentelemetry-4.11.0	org.axonframework.tracing.opentelemetry	io.opentelemetry.context.propagation	9	
22	axon-tracing-opentelemetry-4.11.0	org.axonframework.tracing.opentelemetry	io.opentelemetry.api.trace	9	
23	axon-eventsourcing-4.11.0	org.axonframework.eventsourcing.eventstore.leg...	org.slf4j	8	
24	axon-messaging-4.11.0	org.axonframework.common.caching	org.ehcache.event	8	
25	axon-messaging-4.11.0	org.axonframework.common.caching	javax.cache.event	8	
26	axon-messaging-4.11.0	org.axonframework.messaging.annotation	org.slf4j	8	

	artifactName	fullPackageName	externalPackageName	numberOfExternalTypeCaller	numberOfExt
27	axon-server-connector-4.11.0	org.axonframework.axonserver.connector	io.axoniq.axonserver.grpc	8	
28	axon-server-connector-4.11.0	org.axonframework.axonserver.connector.command	io.axoniq.axonserver.grpc	8	
29	axon-server-connector-4.11.0	org.axonframework.axonserver.connector.event.axon	io.axoniq.axonserver.grpc.event	8	
30	axon-server-connector-4.11.0	org.axonframework.axonserver.connector.util	org.slf4j	8	
31	axon-test-4.11.0	org.axonframework.test.saga	org.hamcrest	8	
32	axon-messaging-4.11.0	org.axonframework.messaging.responsetypes	reactor.core.publisher	7	
33	axon-messaging-4.11.0	org.axonframework.queryhandling	org.reactivestreams	7	
34	axon-messaging-4.11.0	org.axonframework.serialization.avro	org.apache.avro.message	7	
35	axon-messaging-4.11.0	org.axonframework.serialization.avro	org.apache.avro.generic	7	
36	axon-server-connector-4.11.0	org.axonframework.axonserver.connector.event.axon	io.axoniq.axonserver.grpc	7	
37	axon-server-connector-4.11.0	org.axonframework.axonserver.connector.query	reactor.core.publisher	7	
38	axon-server-connector-4.11.0	org.axonframework.axonserver.connector.query.s...	reactor.core.publisher	7	
39	axon-eventsourcing-4.11.0	org.axonframework.eventsourcing.eventstore	org.slf4j	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	nu
0	axon-spring-boot-autoconfigure-4.11.0	org.axonframework.springboot.autoconfigure	AxonAutoConfiguration	20	
1	axon-spring-boot-autoconfigure-4.11.0	org.axonframework.springboot.autoconfigure	AxonServerAutoConfiguration	15	
2	axon-spring-boot-autoconfigure-4.11.0	org.axonframework.springboot.autoconfigure	InfraConfiguration	18	
3	axon-server-connector-4.11.0	org.axonframework.axonserver.connector.query	AxonServerQueryBus	15	
4	axon-spring-boot-autoconfigure-4.11.0	org.axonframework.springboot.autoconfigure	PersistentStreamMessageSourceRegistrar	13	
5	axon-spring-boot-autoconfigure-4.11.0	org.axonframework.springboot.util	AbstractQualifiedBeanCondition	12	
6	axon-server-connector-4.11.0	org.axonframework.axonserver.connector.event.axon	AxonServerEventStore\$AxonIQEventStorageEngine	12	
7	axon-spring-boot-autoconfigure-4.11.0	org.axonframework.springboot.autoconfigure	XStreamAutoConfiguration	11	
8	axon-spring-boot-autoconfigure-4.11.0	org.axonframework.springboot.autoconfigure	AxonDbSchedulerAutoConfiguration	9	
9	axon-spring-boot-autoconfigure-4.11.0	org.axonframework.springboot.autoconfigure	JpaAutoConfiguration	10	
10	axon-spring-boot-autoconfigure-4.11.0	org.axonframework.springboot.autoconfigure	AvroSerializerAutoConfiguration	13	
11	axon-spring-boot-autoconfigure-4.11.0	org.axonframework.springboot.autoconfigure	MicrometerMetricsAutoConfiguration	13	
12	axon-spring-boot-autoconfigure-4.11.0	org.axonframework.springboot.autoconfigure.legacyjpa	JpaJavaxAutoConfiguration	11	
13	axon-messaging-4.11.0	org.axonframework.serialization.json	JacksonSerializer	9	
14	axon-server-connector-4.11.0	org.axonframework.axonserver.connector	AxonServerConnectionManager\$Builder	9	
15	axon-server-connector-4.11.0	org.axonframework.axonserver.connector.event.axon	AxonServerEventScheduler	10	
16	axon-server-connector-4.11.0	org.axonframework.axonserver.connector.processor	EventProcessorControlService	8	
17	axon-server-connector-4.11.0	org.axonframework.axonserver.connector.query	QueryProcessingTask	10	
18	axon-spring-boot-autoconfigure-4.11.0	org.axonframework.springboot.autoconfigure	JdbcAutoConfiguration	9	
19	axon-spring-boot-autoconfigure-4.11.0	org.axonframework.springboot.autoconfigure	MetricsAutoConfiguration	11	

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	p
0	axon-messaging-4.11.0	66	809	49	46	167	20.642769	
1	axon-server-connector-4.11.0	11	142	24	10	88	61.971831	
2	axon-spring-boot-autoconfigure-4.11.0	9	88	47	8	39	44.318182	
3	axon-modelling-4.11.0	10	158	3	7	12	7.594937	
4	axon-test-4.11.0	8	87	12	6	36	41.379310	
5	axon-eventsourcing-4.11.0	9	133	3	5	15	11.278195	
6	axon-configuration-4.11.0	1	41	1	1	6	14.634146	
7	axon-disruptor-4.11.0	1	22	4	1	10	45.454545	
8	axon-tracing-opentelemetry-4.11.0	1	5	5	1	5	100.000000	

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-4.11.0	axon-messaging-4.11.0	axon-modelling-4.11.0	axon-server-connector-4.11.0	axon-spring-boot-autoconfigure-4.11.0	axon-test-4.11.0
numberOfPackages						
2	22.222222	3.030303	20.0	18.181818	22.222222	25.0
3	33.333333	0.000000	30.0	27.272727	33.333333	0.0
4	0.000000	6.060606	40.0	36.363636	44.444444	50.0
5	55.555556	0.000000	0.0	0.000000	0.000000	62.5
6	66.666667	0.000000	60.0	54.545455	0.000000	0.0
7	0.000000	10.606061	0.0	63.636364	0.000000	0.0
8	0.000000	0.000000	0.0	72.727273	0.000000	0.0
9	0.000000	0.000000	0.0	81.818182	0.000000	0.0
11	0.000000	16.666667	0.0	0.000000	0.000000	0.0
39	0.000000	59.090909	0.0	0.000000	0.000000	0.0
49	0.000000	74.242424	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	maxNi
0	axon-messaging-4.11.0	66	49	1	1.0	2.102041	
1	axon-server-connector-4.11.0	11	24	1	2.0	2.500000	
2	axon-modelling-4.11.0	10	3	2	3.0	3.666667	
3	axon-event sourcing-4.11.0	9	3	1	2.0	2.666667	
4	axon-test-4.11.0	8	12	1	1.0	1.416667	
5	axon-spring-boot-autoconfigure-4.11.0	9	47	1	1.0	1.276596	
6	axon-configuration-4.11.0	1	1	1	1.0	1.000000	
7	axon-disruptor-4.11.0	1	4	1	1.0	1.000000	
8	axon-tracing-opentelemetry-4.11.0	1	5	1	1.0	1.000000	

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

	artifactName	artifactPackages	numberOfExternalPackages	minNumberOfPackagesPercentage	medNumberOfPackagesPercentage	avgNum
0	axon-messaging-4.11.0	66	49	1.515152	1.515152	
1	axon-server-connector-4.11.0	11	24	9.090909	18.181818	
2	axon-modelling-4.11.0	10	3	20.000000	30.000000	
3	axon-eventsourcing-4.11.0	9	3	11.111111	22.222222	
4	axon-test-4.11.0	8	12	12.500000	12.500000	
5	axon-spring-boot-autoconfigure-4.11.0	9	47	11.111111	11.111111	
6	axon-configuration-4.11.0	1	1	100.000000	100.000000	
7	axon-disruptor-4.11.0	1	4	100.000000	100.000000	
8	axon-tracing-opentelemetry-4.11.0	1	5	100.000000	100.000000	

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

	artifactName	artifactTypes	numberOfExternalPackages	minNumberOfTypes	medNumberOfTypes	avgNumberOfTypes	maxNumberOfTypes
0	axon-messaging-4.11.0	809	49	1	2.0	5.020408	82
1	axon-server-connector-4.11.0	142	24	1	4.0	7.791667	30
2	axon-modelling-4.11.0	158	3	3	3.0	5.000000	9
3	axon-eventsourcing-4.11.0	133	3	3	3.0	6.666667	14
4	axon-test-4.11.0	87	12	1	1.5	3.833333	27
5	axon-spring-boot-autoconfigure-4.11.0	88	47	1	1.0	1.957447	6
6	axon-configuration-4.11.0	41	1	6	6.0	6.000000	6
7	axon-disruptor-4.11.0	22	4	1	5.5	5.000000	8
8	axon-tracing-opentelemetry-4.11.0	5	5	1	2.0	2.200000	4

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

	artifactName	artifactTypes	numberOfExternalPackages	minNumberOfTypesPercentage	medNumberOfTypesPercentage	avgNumberOfTypes
0	axon-messaging-4.11.0	809	49	0.123609	0.247219	
1	axon-server-connector-4.11.0	142	24	0.704225	2.816901	
2	axon-modelling-4.11.0	158	3	1.898734	1.898734	
3	axon-eventsourcing-4.11.0	133	3	2.255639	2.255639	
4	axon-test-4.11.0	87	12	1.149425	1.724138	
5	axon-spring-boot-autoconfigure-4.11.0	88	47	1.136364	1.136364	
6	axon-configuration-4.11.0	41	1	14.634146	14.634146	
7	axon-disruptor-4.11.0	22	4	4.545455	25.000000	
8	axon-tracing-opentelemetry-4.11.0	5	5	20.000000	40.000000	

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>

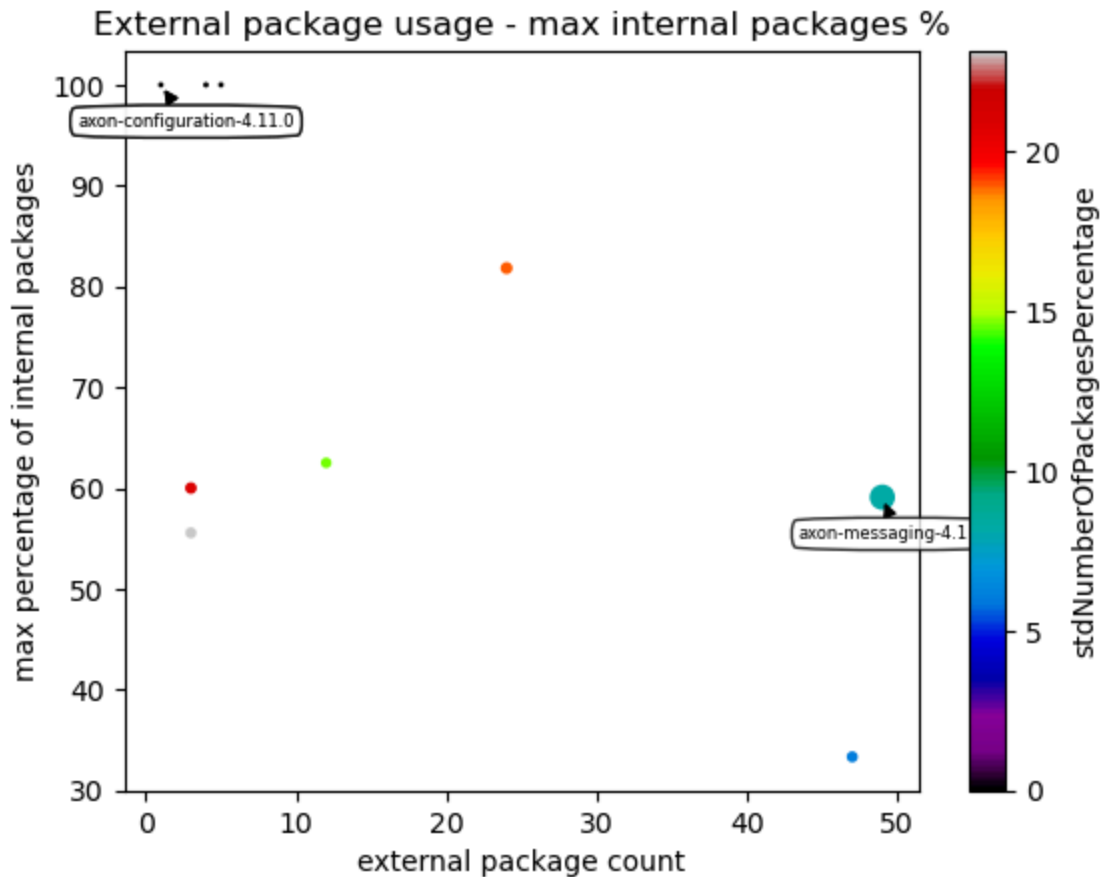
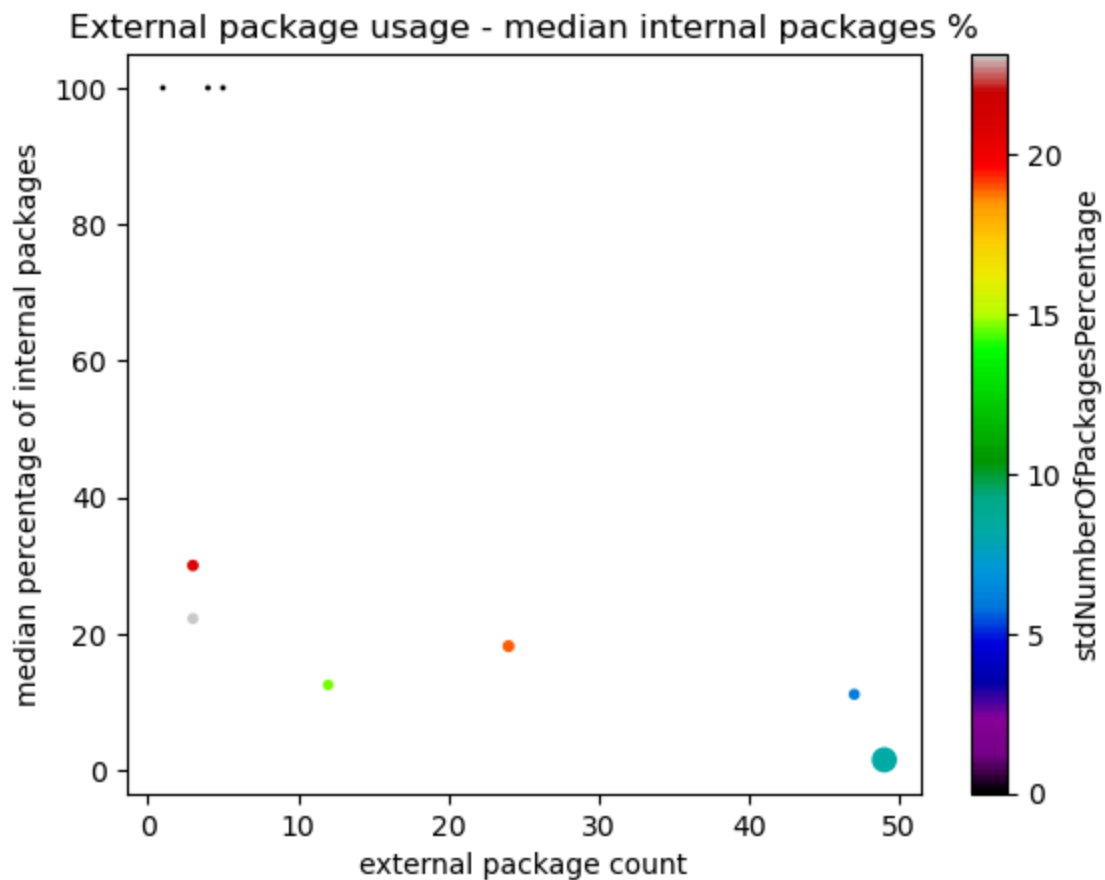


Table 13 Chart 2 - External package usage - median percentage of internal types

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	Axon Framework - Configuration	test	False	jakarta.persistence	jakarta.persistence-api
1	axon-configuration	Axon Framework - Configuration	default	False	org.axonframework	axon-modelling
2	axon-configuration	Axon Framework - Configuration	test	False	org.quartz-scheduler	quartz
3	axon-configuration	Axon Framework - Configuration	test	False	org.hibernate	hibernate-core-jakarta
4	axon-configuration	Axon Framework - Configuration	default	False	\${project.groupId}	axon-eventsourcing
...
160	axon-test	Axon Framework - Test Fixtures	test	False	jakarta.persistence	jakarta.persistence-api
161	axon-tracing-opentelemetry	Axon Framework - OpenTelemetry Tracing	default	False	io.opentelemetry	opentelemetry-api
162	axon-tracing-opentelemetry	Axon Framework - OpenTelemetry Tracing	provided	False	com.google.code.findbugs	jsr305
163	axon-tracing-opentelemetry	Axon Framework - OpenTelemetry Tracing	default	False	\${project.groupId}	axon-configuration
164	axon-tracing-opentelemetry	Axon Framework - OpenTelemetry Tracing	default	False	\${project.groupId}	axon-messaging

165 rows × 6 columns