

项目申请书

项目名称: Seata Client SDK侧的metrics统计

项目主导师: 刘洋

申请人: 刘戎

日期: 2022.06.23

邮箱: 767817253@qq.com

1. 项目背景

1.1 需求描述

metrics统计有助于帮助使用对Seata的执行过程进行有效的监控，帮助发现可能存在的问题。

1.2 任务

1. 客户端SDK侧的主要监控指标包括：Global begin/commit/rollback， Branch registry/flush undolog/report/commit/rollback等状态统计。
2. 另外，对undolog， fence等hook的调用也要做metrics统计。
3. 现阶段Seata导出的Metrics仅包含服务端相关指标，结合现有Metrics设计，完善Seata client SDK侧的各项指标设计，主要指标如下
Global begin/commit/rollback 等状态的统计
Branch registry/flush undolog/report/commit/rollback 等状态的统计
undolog， fence等hook的调用

1.3 Seata介绍

2. 技术方法与可行性

2.1 Seata的Metrics模块分析

2.1.1 Metrics模块设计思路

在Seata官方文档中，开发者提供了Seata的Metrics模块的设计思路如下：

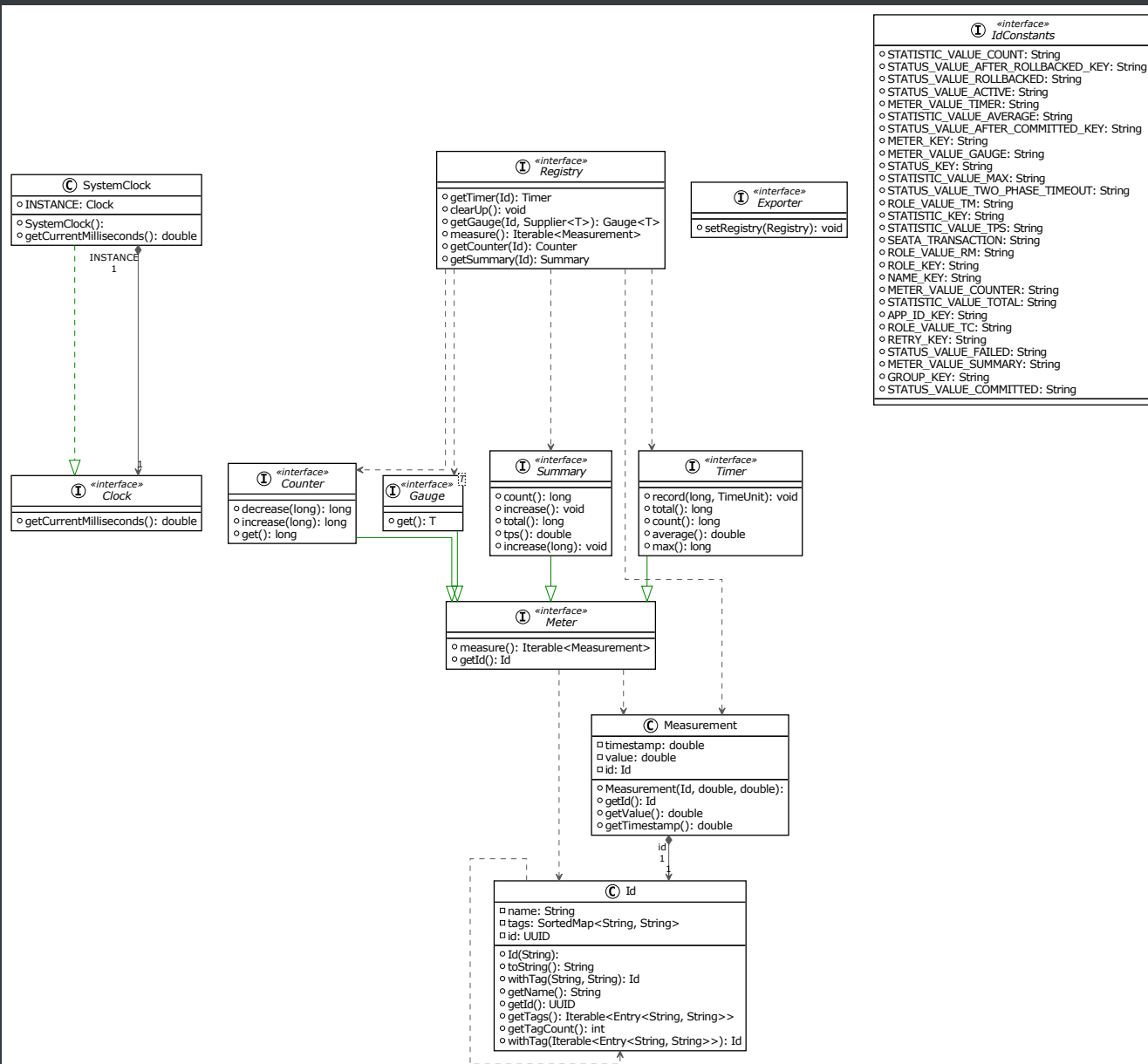
- Seata作为一个被集成的数据一致性框架，Metrics模块将尽可能少的使用第三方依赖以降低发生冲突的风险；
- Metrics模块将竭力争取更高的度量性能和更低的资源开销，尽可能降低开启后带来的副作用；
- 不使用Spring，使用SPI(Service Provider Interface)加载扩展；
- 初始仅发布核心Transaction相关指标，之后结合社区的需求，逐步完善运维方面的所有其他指标。

2.1.2 Metrics模块实现

Seata的Metrics模块，由2个核心API模块 `seata-metrics-api` 和 `seata-metrics-core`，以及N个实现模块例如 `seata-metrics-registry-compact`、`seata-metrics-exporter-prometheus` 构成。

- **`seata-metrics-api`模块**

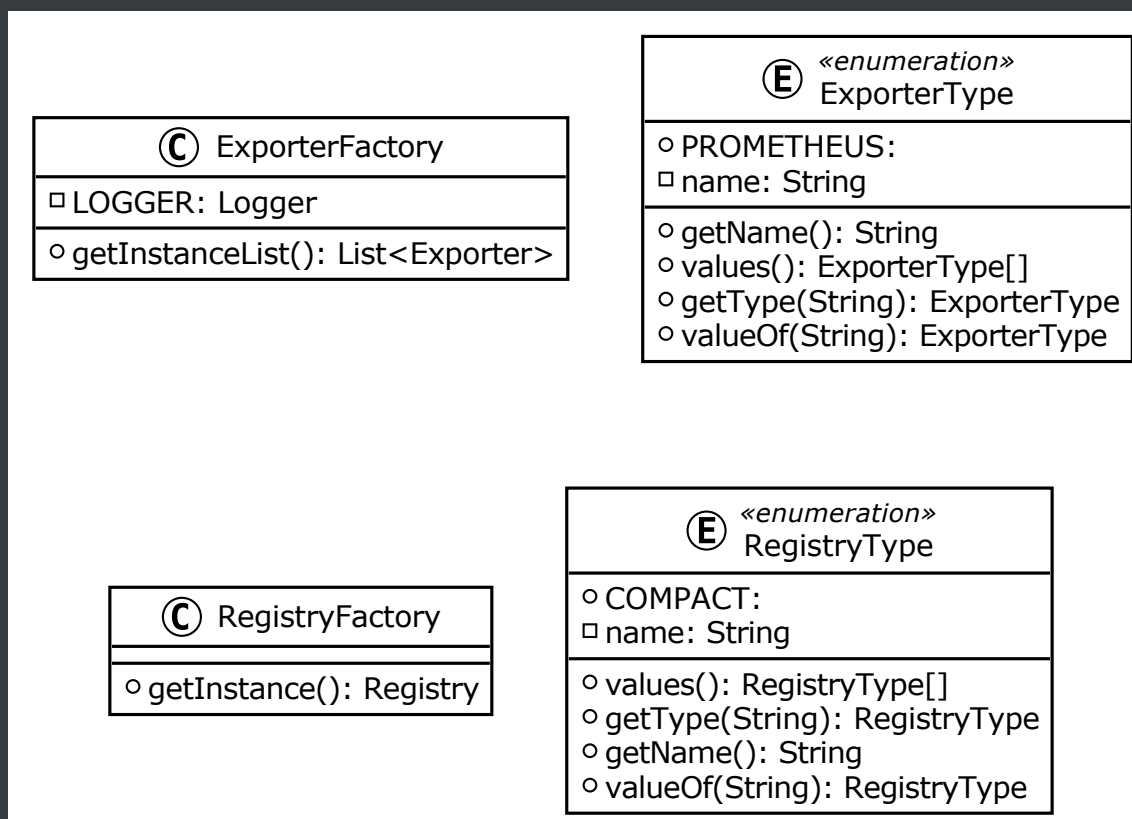
阅读代码可得 `seata-metrics-api` 模块，是Metrics的核心，将作为Seata基础架构的一部分被TC、TM和RM引用，它内部没有任何具体实现代码，仅包含接口定义。模块定义的类与接口如下：



- IDConstants : Metrics模块所使用的常量 String 。
- Id : 作为一个基类，其定义了一个数据指标的元信息：id、name、tags。
- Measurement : 这个类组合了Id类，定义了每个timestamp，metric指标的value。
- Meter : 作为一个父接口，其被Gauge、Counter、Summary、Timer实现，定义了获取每个度量器的measurement数据的方法。
- Counter : 单一累加度量器的接口。
- Gauge : 单一最新值度量器的接口。
- Summary : 多Measurement输出计数器的接口。
- Timer : 多Measurement输出计时器的接口。
- Registry : 定义了获取在registry中注册的各个Meter实现度量器。
- Exporter : 定义了设定registry方法。
- Clock&SystemClock : 用于获取度量器值对应的(时序型数据)timestamp，

■ seata-metrics-core模块

阅读代码可得 seata-metrics-core 模块是Metrics核心模块，根据配置组织（加载）1个Registry和N个Exporter
模块定义的类与接口如下：

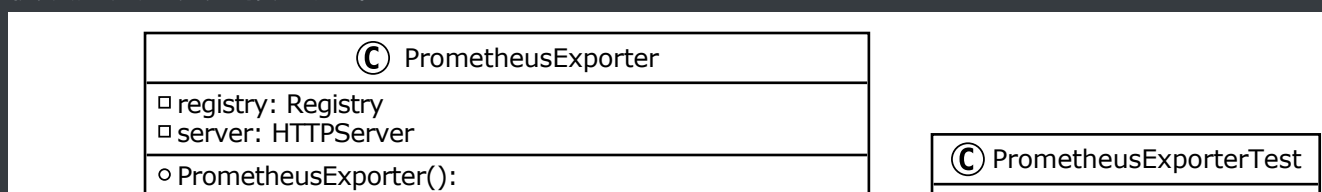


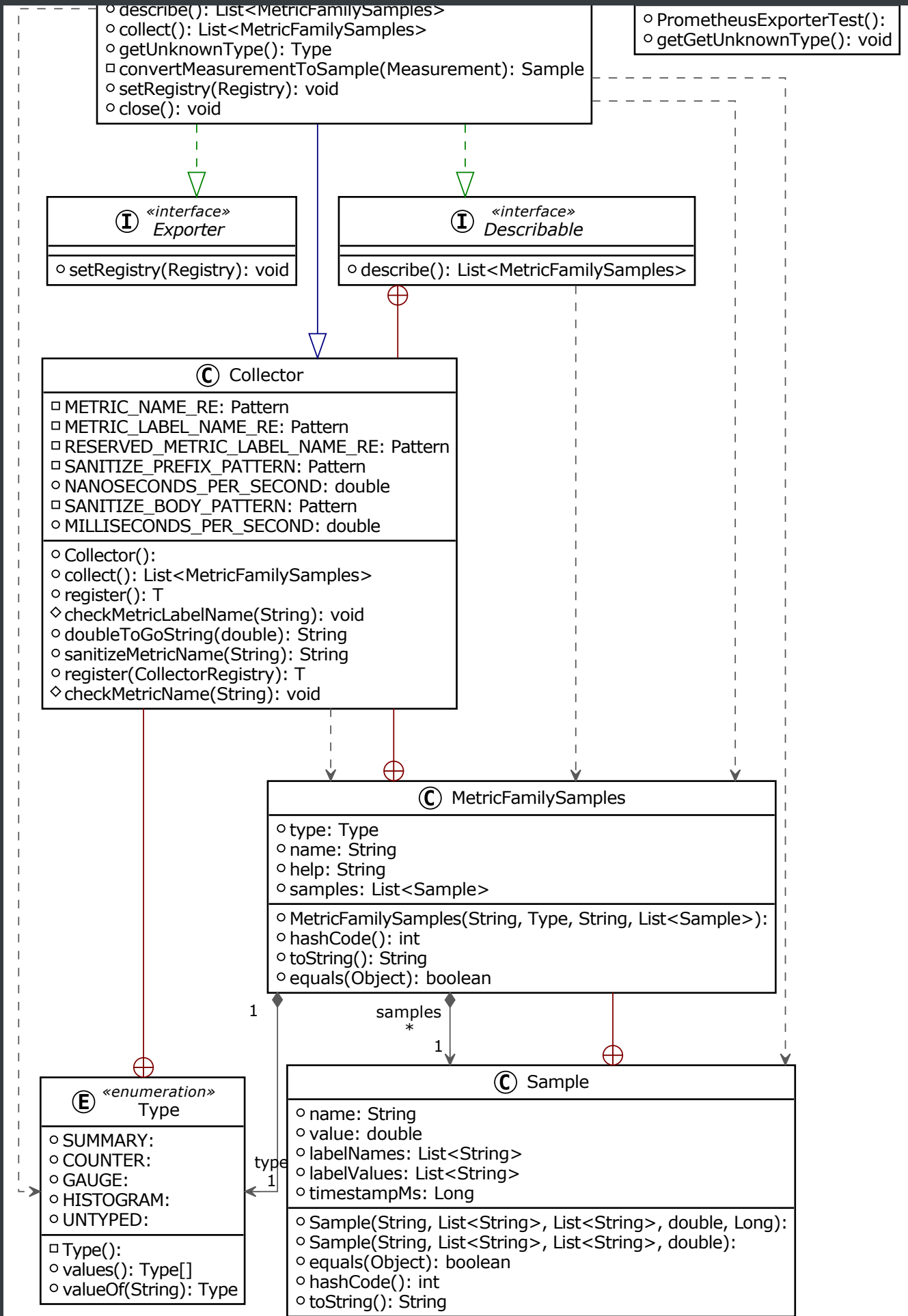
- `ExporterFactory` : 实现了获取所有在配置文件中 `metrics.exporterList` 中配置的所有 exporter 名称，并在基于 seata 内部的 `EnhancedServiceLoader` 与 `InnerEnhancedServiceLoader` 两个类中定义的 SPI 加载机制，加载所有 `Exporter` 单实例到 JVM 中，并最后返回由所有 exporter 的名称构成的 `ArrayList`。
- `ExporterType` : 定义了 Prometheus Exporter 的元数据。
- `RegistryFactory` : 同 `ExporterFactory` 机制类似，加载所有配置文件中的 registry 配置，并创建 & 加载单 registry 实例，
- `RegistryType` : 定义了 Registry 的种类，现在仅实现了 `COMPACT` 类型，后续可扩展。

■ seata-metrics-exporter-prometheus模块

阅读代码可得 seata-metrics-exporter-prometheus 是一个 `PrometheusExporter` 的实现。

模块定义的类与接口如下：

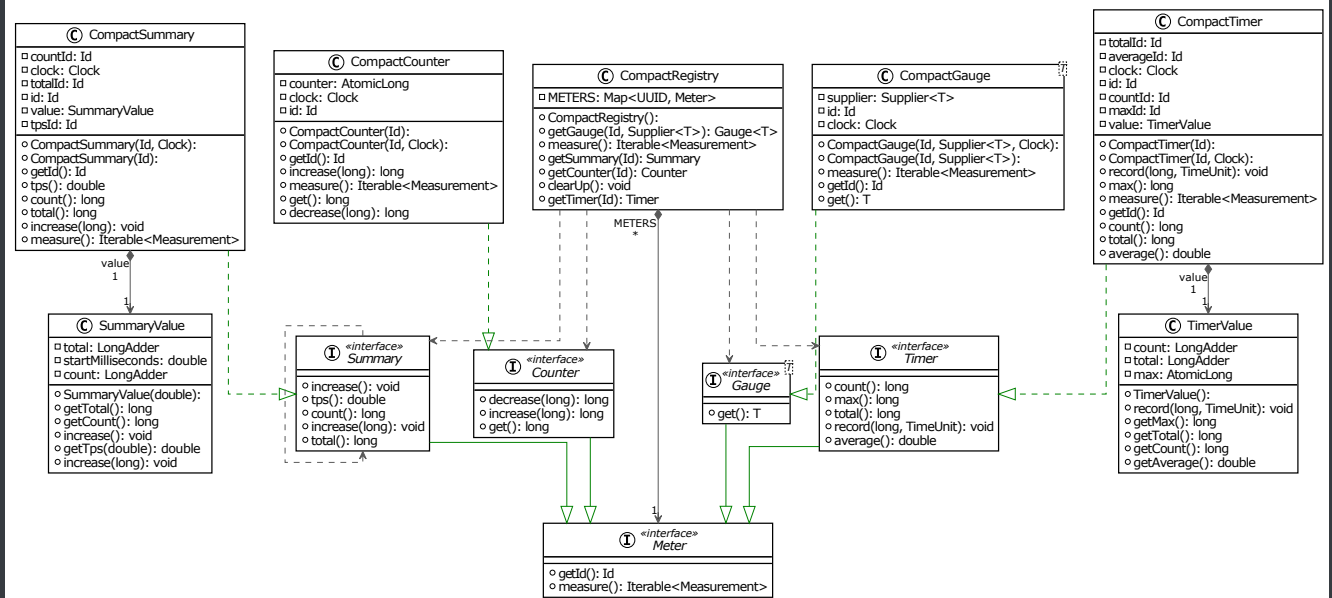




- `seata-metrics-exporter-prometheus` : 这个类继承自 `prometheus-client` 的 `Collector`

类，实现了 Exporter 接口,可自行设定register;采集seata-server侧的metrics指标，转换为prometheus的 Sample 格式的数据;作为服务后端运行，向外暴露采集到的metrics。

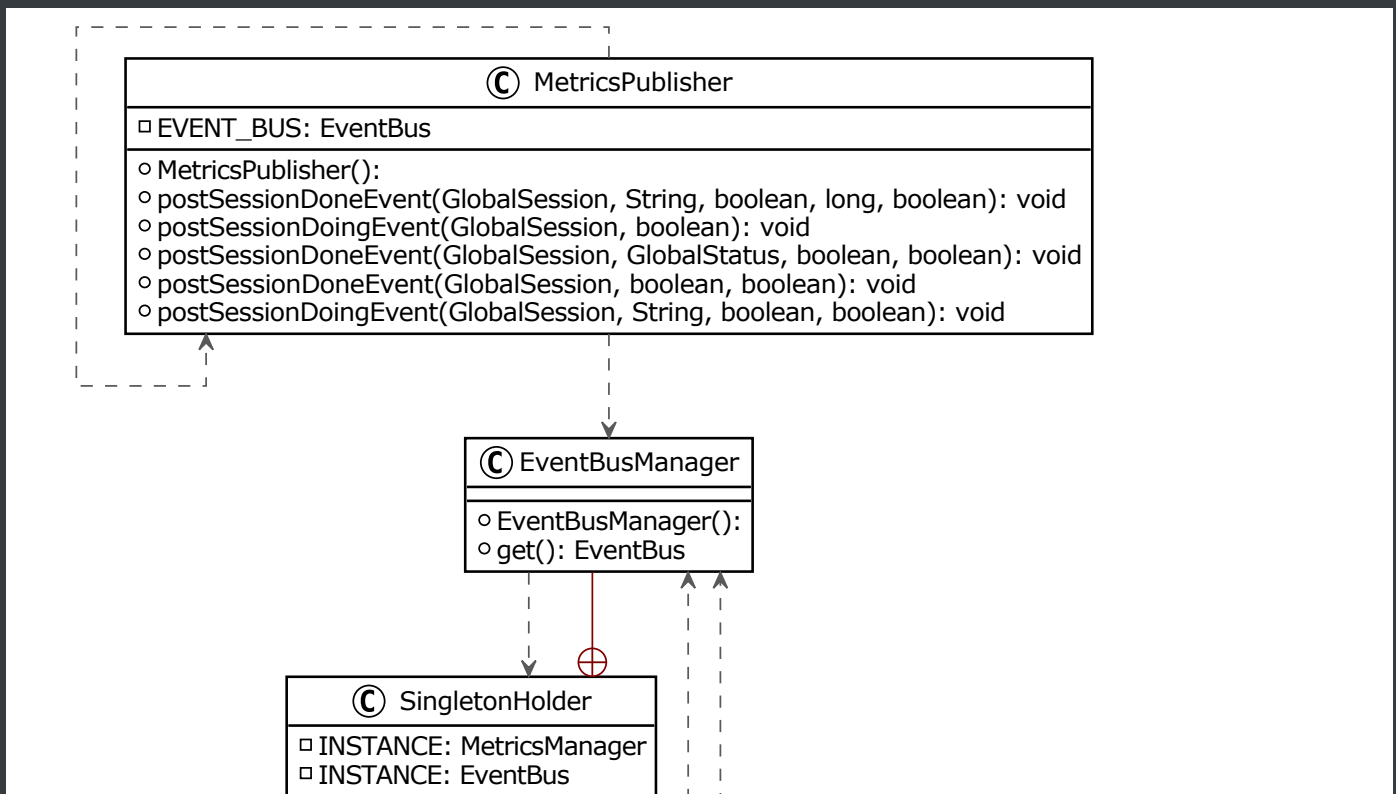
seata-metrics-registry-compact模块

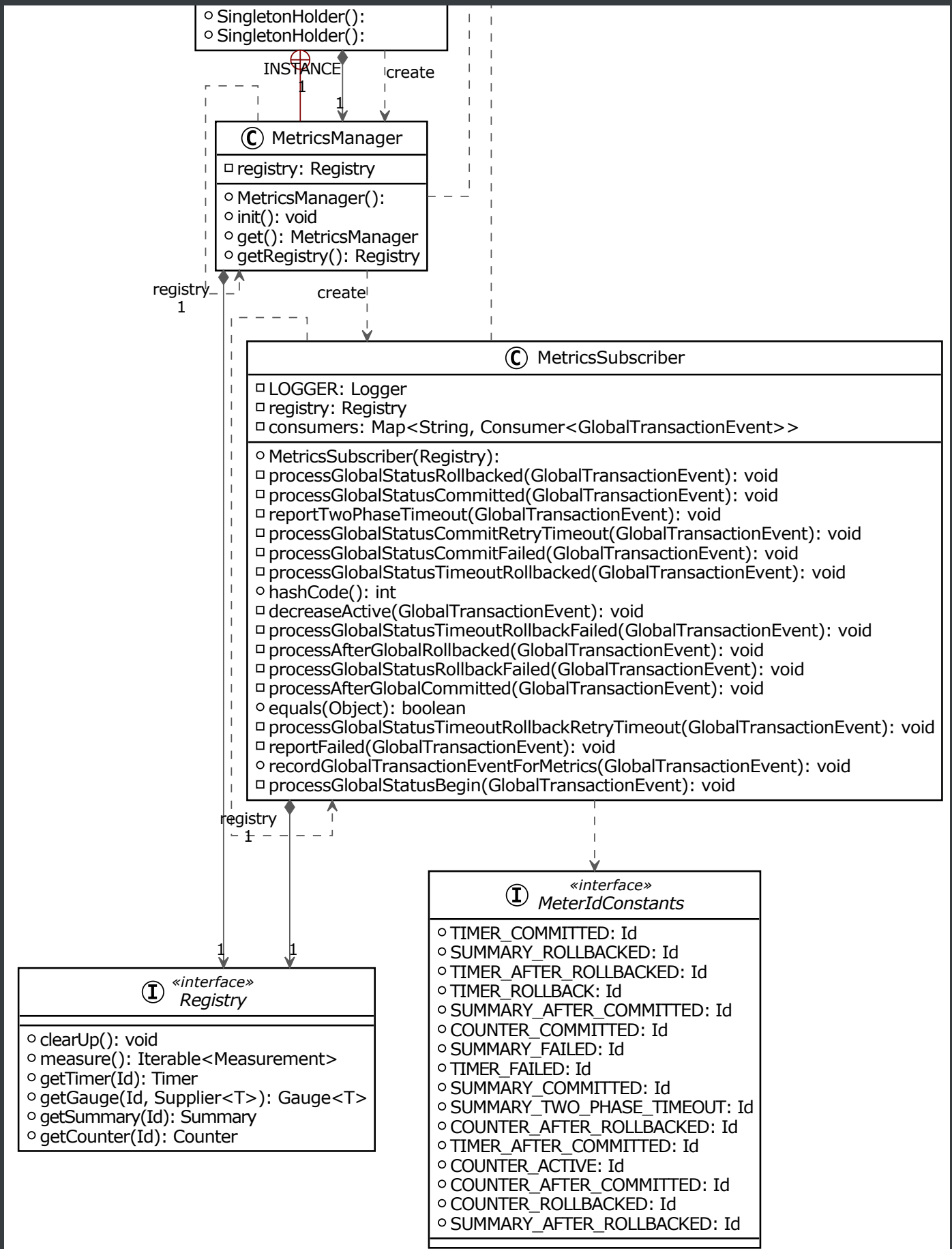


阅读代码可得 seata-metrics-registry-compact 实现了多个Metrics度量器，与一个 CompactRegistry。

2.2 seata-server (TC) 实现的metrics结构分析

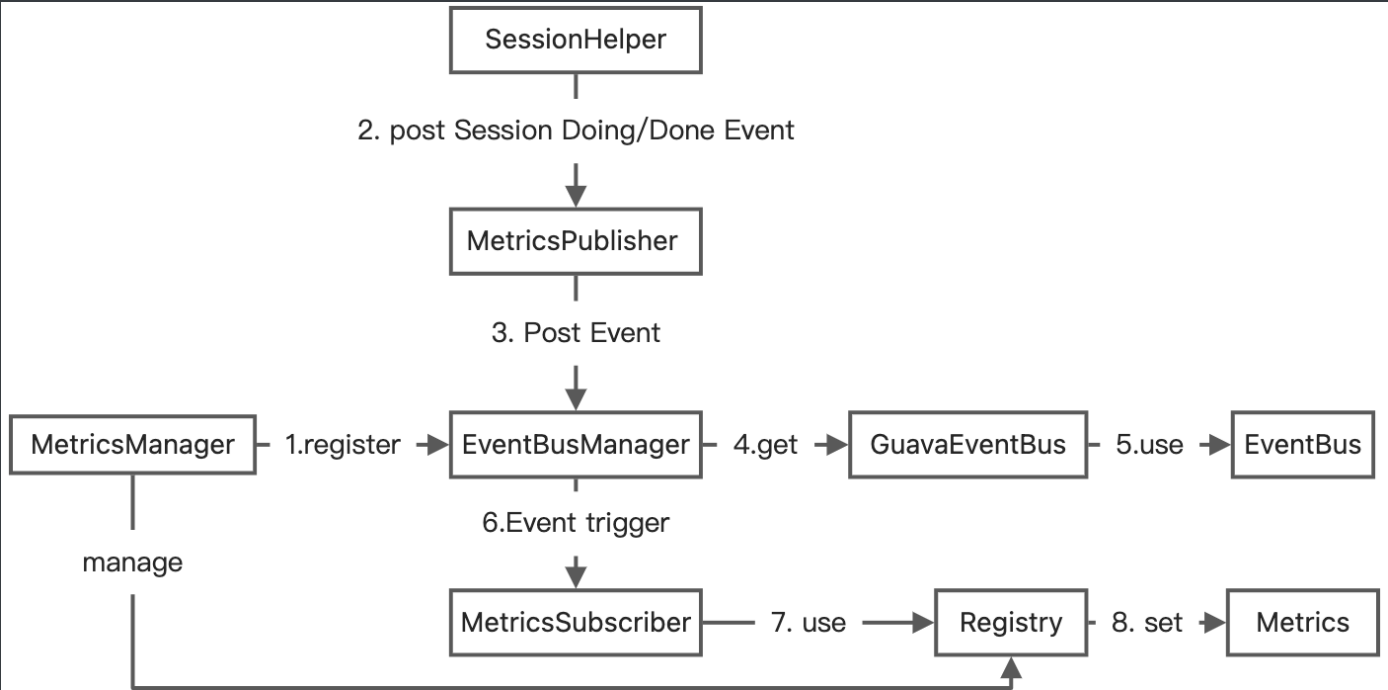
seata-server实现了Metrics。可以看到，这一部分一共由3个类，一个接口组成。





- **MeterIdConstants** : 这个接口定义了多个Id，这些Id中包含特定的tag，对应不同的Meter采集度量指标。

- MetricsPublisher : 这个类主要通过 GlobalSession 实例对象, 向 EventBus (单例) 中发布全局事务的Name、beginTime、ServiceGroup等信息。
- MetricsSubscriber : 这个类有一个 HashMap 成员变量, 包含所有全局事务Status状态对应的回调函数. 当全局事务的生命周期中的状态变更时,触发事件对应的回调函数, 将包含其状态的 GlobalTransactionEvent 信息提取, 更新metrics指标。
- MetricsManager : 这个类是一个单例, 在其初始化过程中, 获取 Registry 实例, 并为每个 Exporter 设定registry;在 EventBusManager (单例) 中注册 MetricsSubscriber (由相同registry构造)。



整理可知，seata-server侧的Metrics实现流程如下：首先 MetricsManager 创建 MetricsSubscriber 实例，并通过 EventBusManager 将其注册到 Registry 中.之后在全局分布式事务的生命周期中埋点，将其生命周期中关键事件通过 SessionHelper 发布到 MetricsPublisher 中，被 EventBusManager 监听到事件后，根据 Session 中包含的事物信息，触发 Metrics Subscriber 中注册的回调函数，通过 Registry 记录含有特定tag的 metric指标。

2.3 seata-server采集的metrics

Metrics	Description
seata.transaction(role=tc,meter=counter,status=active/committed/rollback)	当前活动中 已提交/已回滚 的事务总数

| seata.transaction(role=tc,meter=summary,statistic=count,status=committed/rollback) | 当前周期内 提交/回滚 的事务数 |

| seata.transaction(role=tc,meter=summary,statistic=tps,status=committed/rollback) | 当前周期内 提交/回滚 的事务TPS(transaction per second) |

| seata.transaction(role=tc,meter=timer,statistic=total,status=committed/rollback) | 当前周期内 提交/回滚 的事务耗时总和 |

| seata.transaction(role=tc,meter=timer,statistic=count,status=committed/rollback) | 当前周期内 提交/回滚 的事务数 |

| seata.transaction(role=tc,meter=timer,statistic=average,status=committed/rollback) | 当前周期内 提交/回滚 的事务平均耗时 |

| seata.transaction(role=tc,meter=timer,statistic=max,status=committed/rollback) | 当前周期内 提交/回滚 的事务最大耗时 |

3. 项目实现细节

3.1 其他如Prometheus等开源结构的metrics-server、client是怎么实现的

3.1 seata-client的指标采集架构

类图设计

spi提及

流程图

本设计的性能测试与改进规划

3.2 seata-client侧的metrics

4. 项目进度安排

5. 个人经历

5.1 seata相关

自己的关于seata的实习经历

5.2 指标采集&监控相关

自己关于指标采集&监控的实习经历

本文引用

1. <https://seata.io/zh-cn/docs/dev/seata-mertics.html>