

微服务架构梳理

* 2018微服务架构设计图



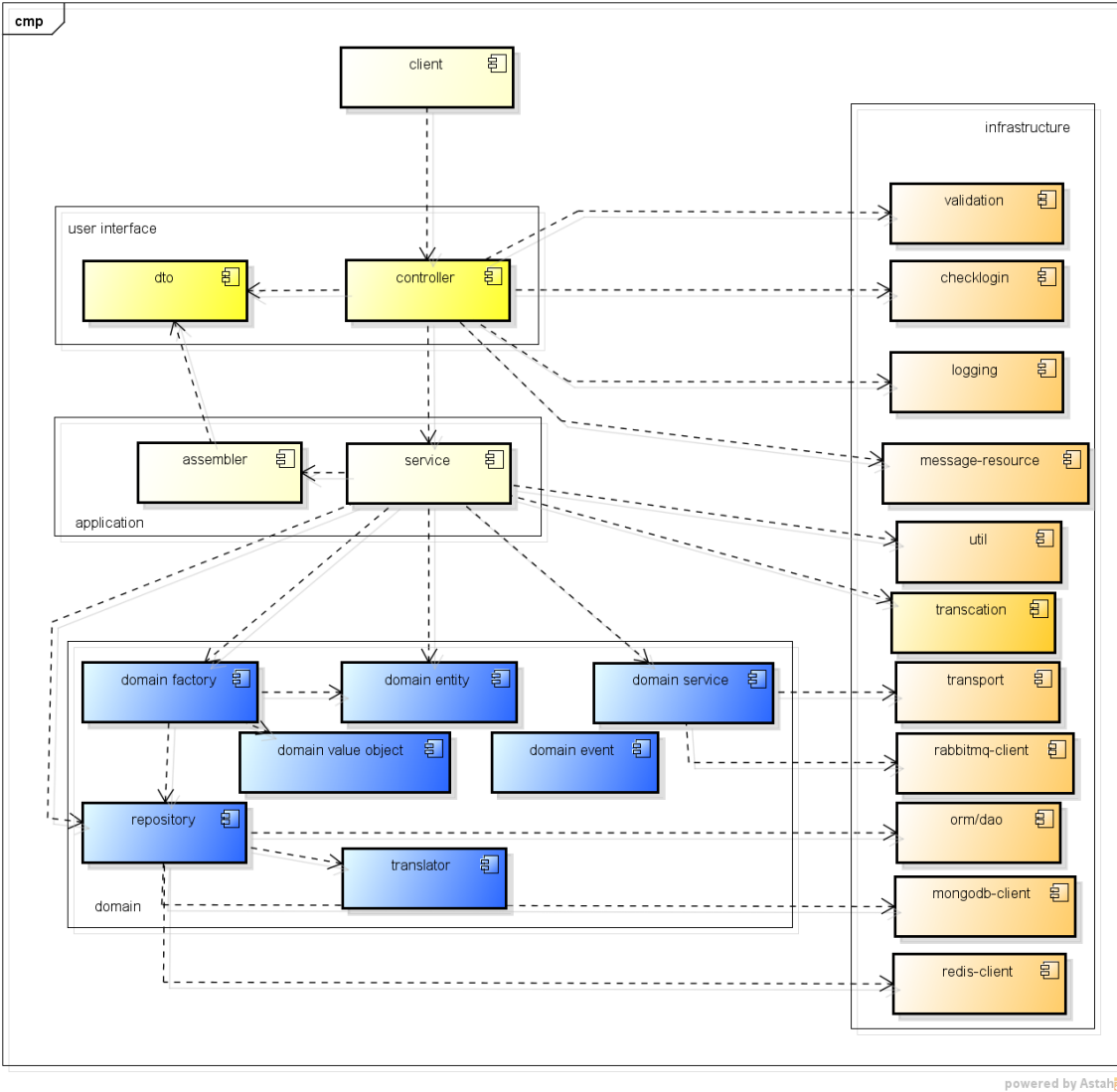
思维：大中台，小前台，轻后台，业务高内聚，数据统一管理。

中台：统一接口规范，采集多个后台数据。
统一设计管理业务流程，企业数据。
小前台，低耦合，多形态，微服务。

适用场景：复杂业务，复杂 IT 产品，业务数据多版本。

- 业务流程复杂，流程跨域流转，数据需要多次内聚
- IT 产品复杂，老系统系统，第三方成熟产品，数据统一抽取使用
- 业务流程，数据统一梳理，合理管控，统一存储数据中心,最终形成有价值的数
- 据
- 小前台,根据现有业务流程，通过微服务，给各类型前台提供业务数据

* DDD项目架构设计图



client:各类型的客服端，浏览器

User Interface

是用户接口层，为用户/调用方提供可访问的接口，我们简称为“UI”层
我们将dto、controller归入了UI层。同时，在UI层中，我们还会去使用infrastructure层中的validation、logging、checkLogin等公共组件完成一些通用的动作。

Application层

Application层主要职责为组装domain层各个组件及基础设施层的公共组件，完成具体的业务服务。Application层可以理解为粘合各个组件的胶水，使得零散的组件组合在一起提供完整的业务服务。在复杂的业务场景下，一个业务case通常需要多个domain实体参与进来，所以Application的粘合效用正好有了用武之地。
Application层主要由：service、assembler组成，

Domain层

Domain层是具体的业务领域层，是发生业务变化最为频繁的地方，是业务系统最核心的一层，是DDD关注的焦点和难点。这一层包含了如下一些domain object：entity、value object、domain event、domain service、factory、repository等。DDD实践的难点其实就在于如何识别这些object。

infrastructure层

基础设施层提供公共功能组件，供controller、service、domain层调用。