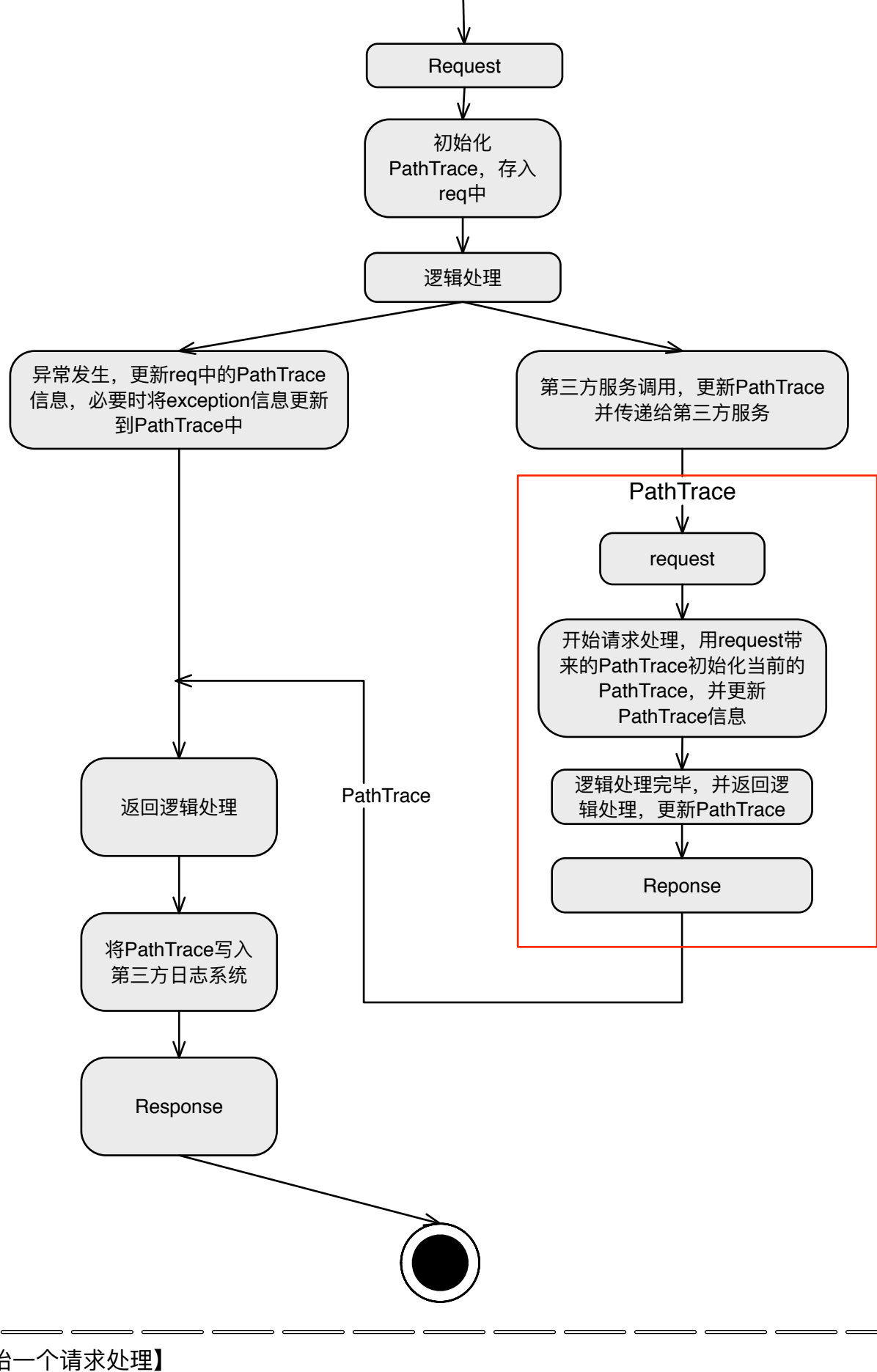
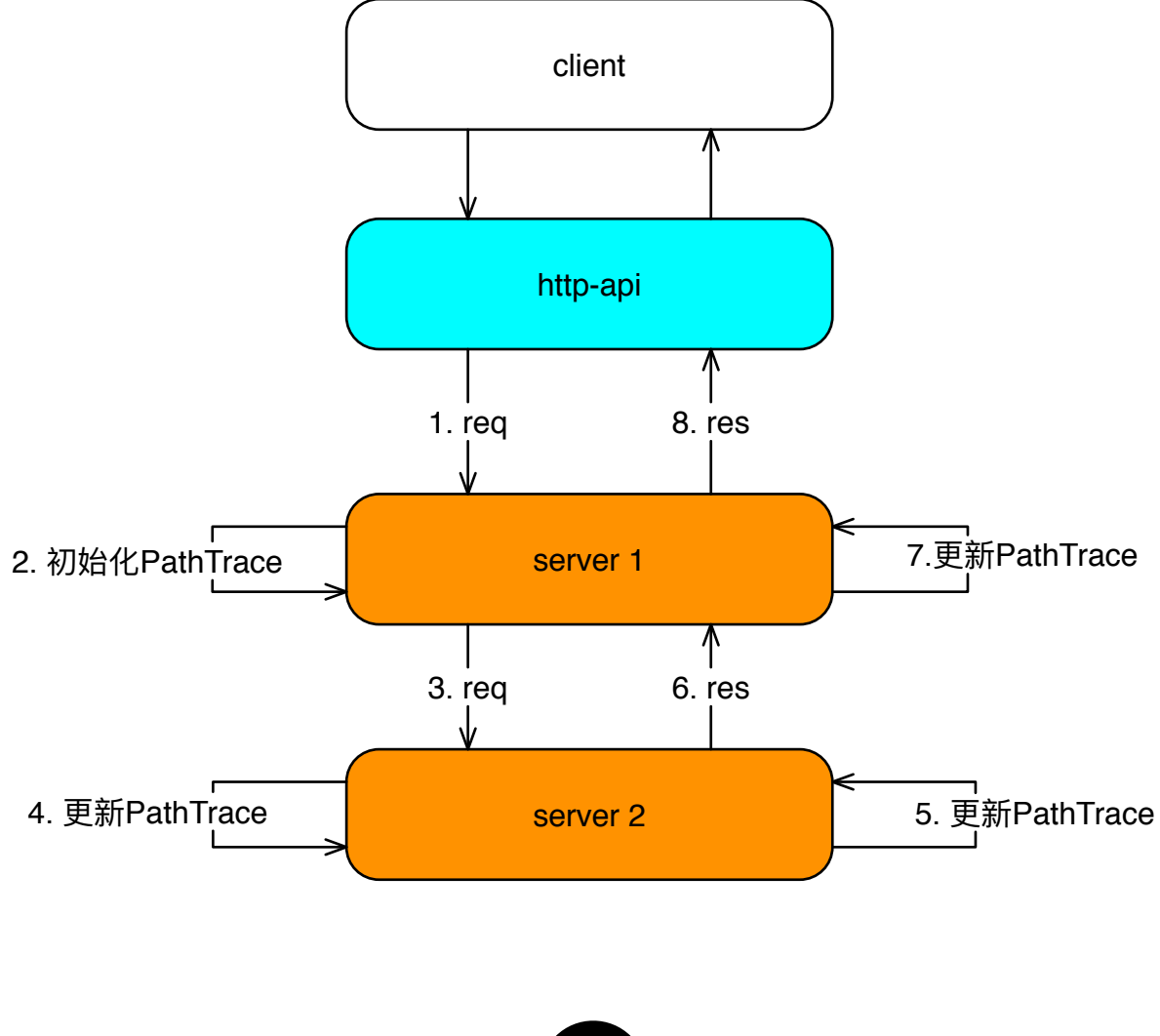


# PathTracer



【开始一个请求处理】  
开始一个请求处理时，如果没有PathTrace的情况下，需要按需初始化一个PathTrace对象，并存入到req。  
【请求第三方服务】  
请求第三方服务时，需要将PathTrace一并传递给第三方服务器。  
【第三方服务处理】  
第三方服务收到请求处理时，用请求带来的PathTrace来初始化自身的PathTrace。  
【结束点的Reponse处理】  
在结束点进行Reponse处理是，如果此条链路是一个异常case的链路，需要将PathTrace写入到日志系统中。

## PathTracer的结构

```
pathTracer:
{
  id      // [string] 当前trace的标识ID，每一个path trace都有独一的ID。
  type    // [string] 标识当前pathtrace的类型。
  date    // [string] 开始处理请求的时间戳。
  router  // [string] 客户端发起请求的api路由。
  pre_path // [array] 在开路的path tracer中，pre_path保存前一路的栈信息。
  path    // [array] 用于记录每一次操作的行为。
}

type: closed(default) | open

// 如果在服务调用过程中，存在不用返回的业务。在调用点将type置为open，成为开路。那么，closed为闭路。

pre_path: [[...], [...], ...]

// 仅当前path trace为开路时才有值。

path:
[
  {sign: "begin",    service_name: "service_1", exception: "", extra: ""},
  {sign: "req-start", service_name: "service_1", router: "", extra: ""},
  {sign: "start",    service_name: "service_2", exception: "", extra: ""},
  {sign: "finish",   service_name: "service_2", exception: "", extra: ""},
  {sign: "req-finish", service_name: "service_1", router: "", extra: ""},
  {sign: "end",      service_name: "service_1", exception: "", extra: ""}
]

{
  sign: "begin | end | req-start | req-finish | start | finish",
  // 表示符号。表示链路中的节点类型。

  service_name: "service_1",
  // 当前行为所处的服务的值。（每个服务都应该有一个唯一的值）

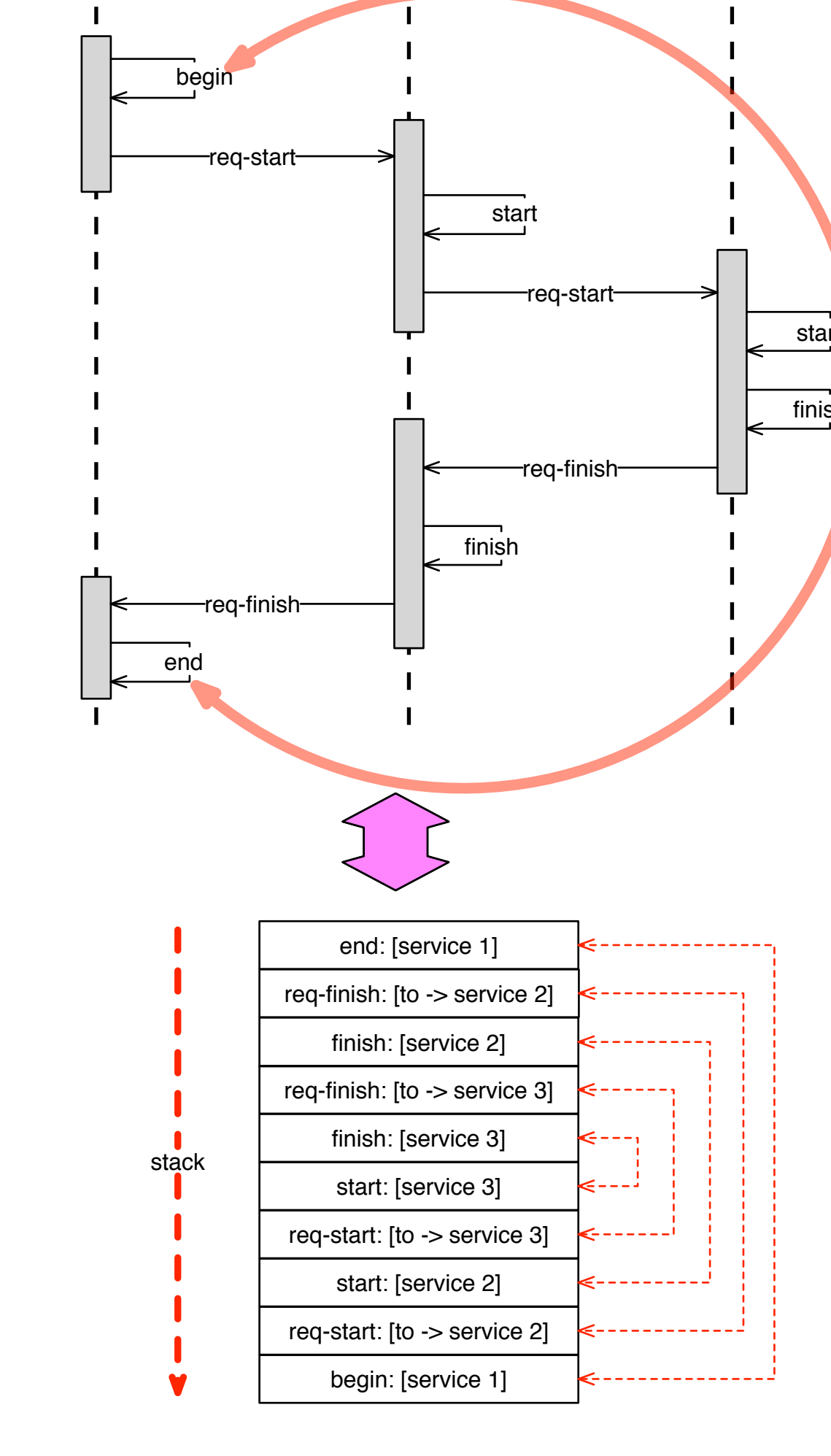
  router: "v3/api/xxx",
  // 记录跨服务的api

  exception: "",
  // 当前服务发生异常的信息。

  extra: ""
  // 放一些额外的信息。
}
```

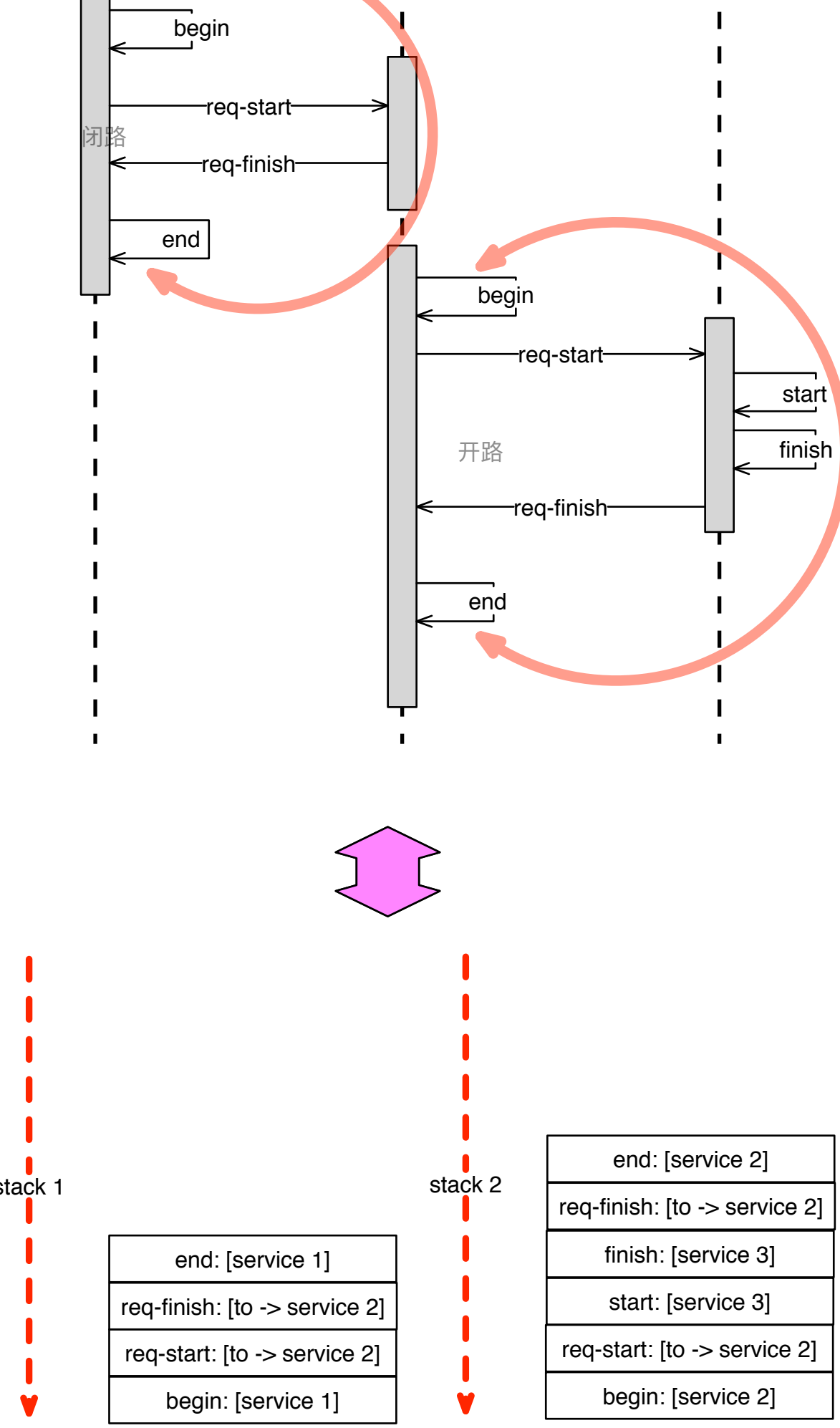
## 闭路 / 开路

【闭路】  
是指服务从收到请求开始，无论中间需要依赖多少次其他服务（或嵌套多少次其他服务），均会产生统一的回路，并从起始服务处处理response。  
闭路path trace能非常完整的收集到整个链路的服务调用栈信息。



- 特点：
1. 一条path trace只有一对begin - end。begin代表最开始，end代表结束。
  2. start - end 代表服务处理当前收到的请求。
  3. reqstart - reqfinish 代表当前服务的逻辑需要依赖其他服务。
  4. 一条path trace可以包含多条start - finish和reqstart - reqfinish，而且可以相互嵌套。

【开路】  
是指服务从收到请求开始，中间调用了第三方微服务，并且不需要第三方服务的返回。从而，服务的链路和第三方服务执行的链路形成了两条独立的pathtrace。



1. 开路的调用方式可能产生多条pathtrace路径。
2. 第2条path trace包涵第1条path trace的调用信息。
3. 多条path trace之间通过 同一个id 来相互关联。
4. 其他同闭路的特点。