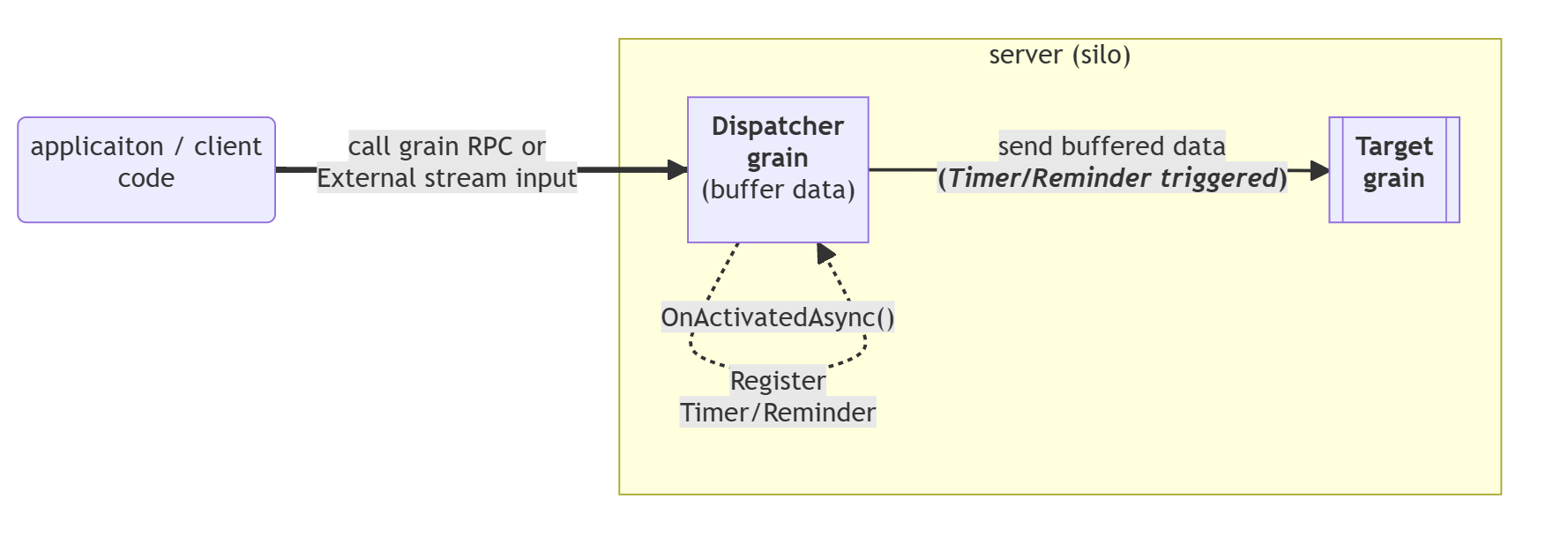
# Orleans常見系統架構模式：Cadence Pattern及其應用範例

## Cadence Pattern 介紹

[Cadence Pattern](https://github.com/OrleansContrib/DesignPatterns/blob/master/Cadence.md) 基本上就是昨天介紹Dispatcher Pattern的一種特定用途衍生型：利用Orleans Grain的Timer/Reminder功能，將一開始接收Client端外部呼叫的Grain做為緩衝儲存區(buffer)，等到設定的Timer/Reminder觸發後，再將該Grain的暫存資料轉送到後端的Grain進行處理：



這個模式的好處是可以將一開始接收到的外部呼叫或是外部資料流量，先暫存起來，累積到一定的資料量或是每逢固定一段時間後，再將資料一次性轉送到後端的Grain進行處理，這樣可以避免後端Grain被頻繁呼叫而拖累系統整體效能，也可以有效的減少後端Grain讀取/儲存狀態的資料庫存取次數，提升後端Grain的處理效能。

由於Grain有Timer/Reminder這種和一般傳統ASP.NET core MVC Controller/Service類別所沒有的特殊功能，也可利用Cadence Pattern來實作一些得做輸入節流閥(throttle)的需求，例如：IoT應用收集設備資料的即時輸入，游戲/直播應用中的訊息對話系統定時禁言；這些需要把短時間從外部湧流進入系統的暫態資料丟棄，只留下最終有意義資訊的功能。

## Cadence Pattern 應用範例：Azure EventHub + Orleans

以下示範如何串接Azure EventHub 輸入的資料，並將資料利用Orleans Stream功能轉送到Cadence Pattern的Orleans後端Grain進行處理。