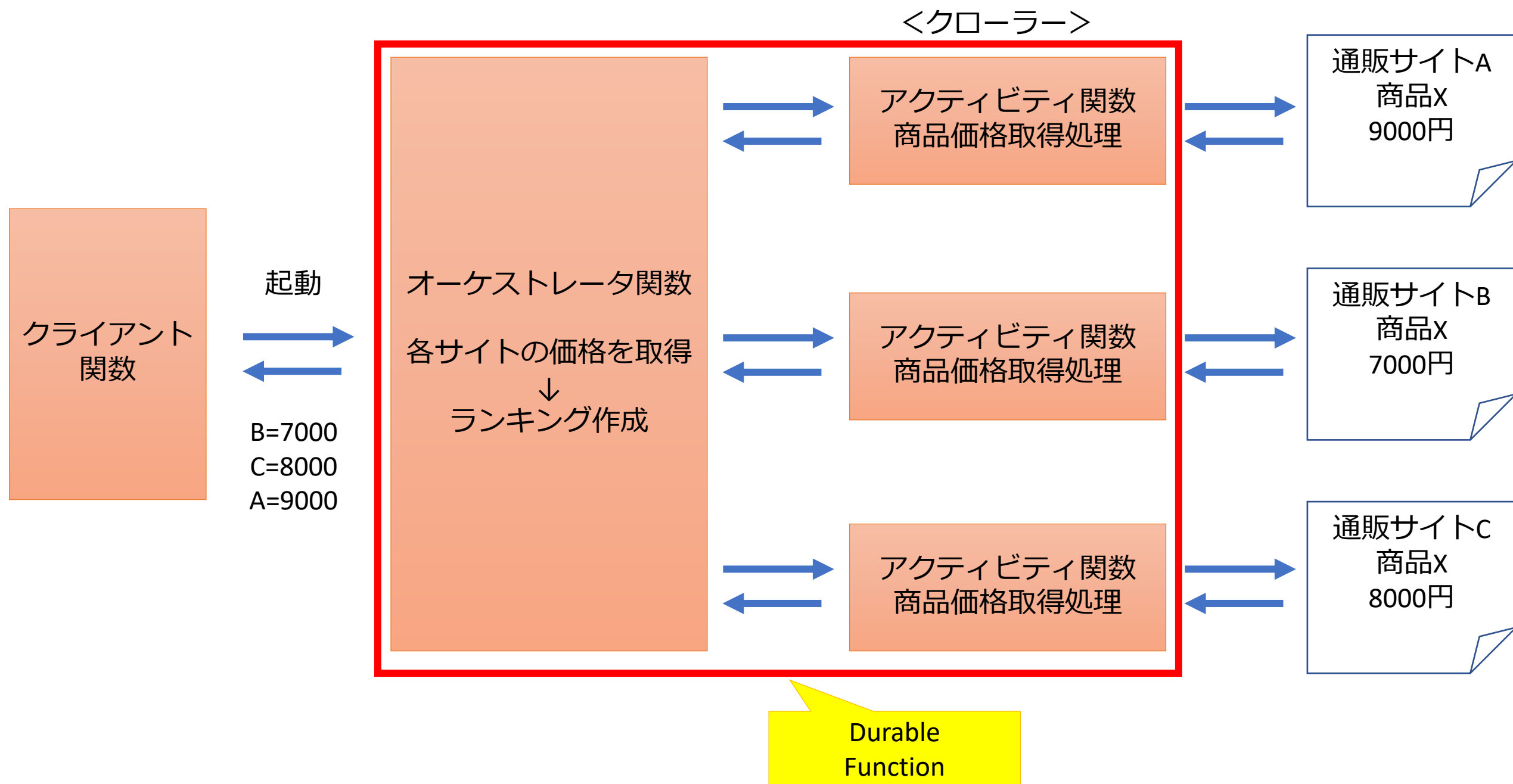


# Durable Functionとは

- Azure Functions の拡張機能
- **ステートフル**
- **長時間**に渡って実行される典型的なワークフローを簡単に実装できる
  - 関数チェーン
  - ファンアウト/ファンイン
  - 非同期HTTP API
  - モニター（状態が変わるまで待機）
  - 人による操作（Human Interaction、承認など）

Durableではない普通のAzure Functionの場合は、実行時間が最長10分（従量課金プランの場合）、ステートレス

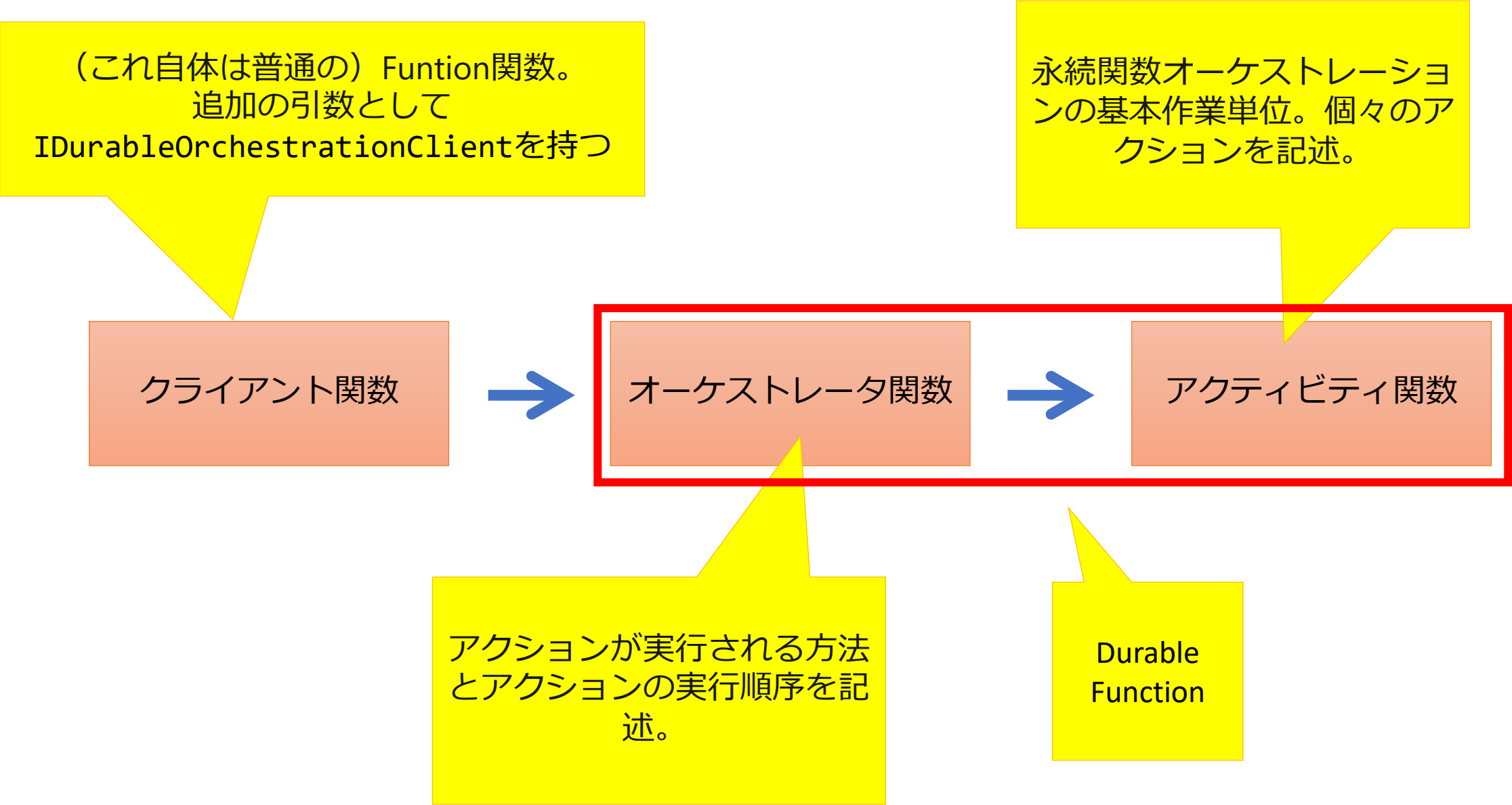
# 「ファンアウト」の例



# サポートされる言語

- C#/C#スクリプト
- JavaScript
- Python
- F#
- PowerShell

3種類の「永続関数」を組み合わせる構成



```
[FunctionName("Durable1_HttpStart")]
public static async Task<HttpResponseMessage> Run(
    [HttpTrigger(...)] HttpRequestMessage req,
    [DurableClient] IDurableOrchestrationClient starter,
    string functionName,
    ILogger log)
{
    await starter.StartNewAsync("Durable1", null);
}
```

```
[FunctionName("Durable1_Hello")]
public static string Run(
    [ActivityTrigger] string name,
    ILogger log)
{
    return $"Hello ${name}";
}
```

クライアント関数



オーケストレータ関数



アクティビティ関数

ふつうの  
Azure  
Function  
関数

```
[FunctionName("Durable1")]
public static async Task<List<string>> RunOrchestrator(
    [OrchestrationTrigger] IDurableOrchestrationContext context )
{
    var outputs = new List<string>();
    outputs.Add( await context.CallFunctionAsync<string>("Durable1_Hello", "Tokyo" );
    outputs.Add( await context.CallFunctionAsync<string>("Durable1_Hello", "Osaka" );
    outputs.Add( await context.CallFunctionAsync<string>("Durable1_Hello", "Nagoya" );
}
```