

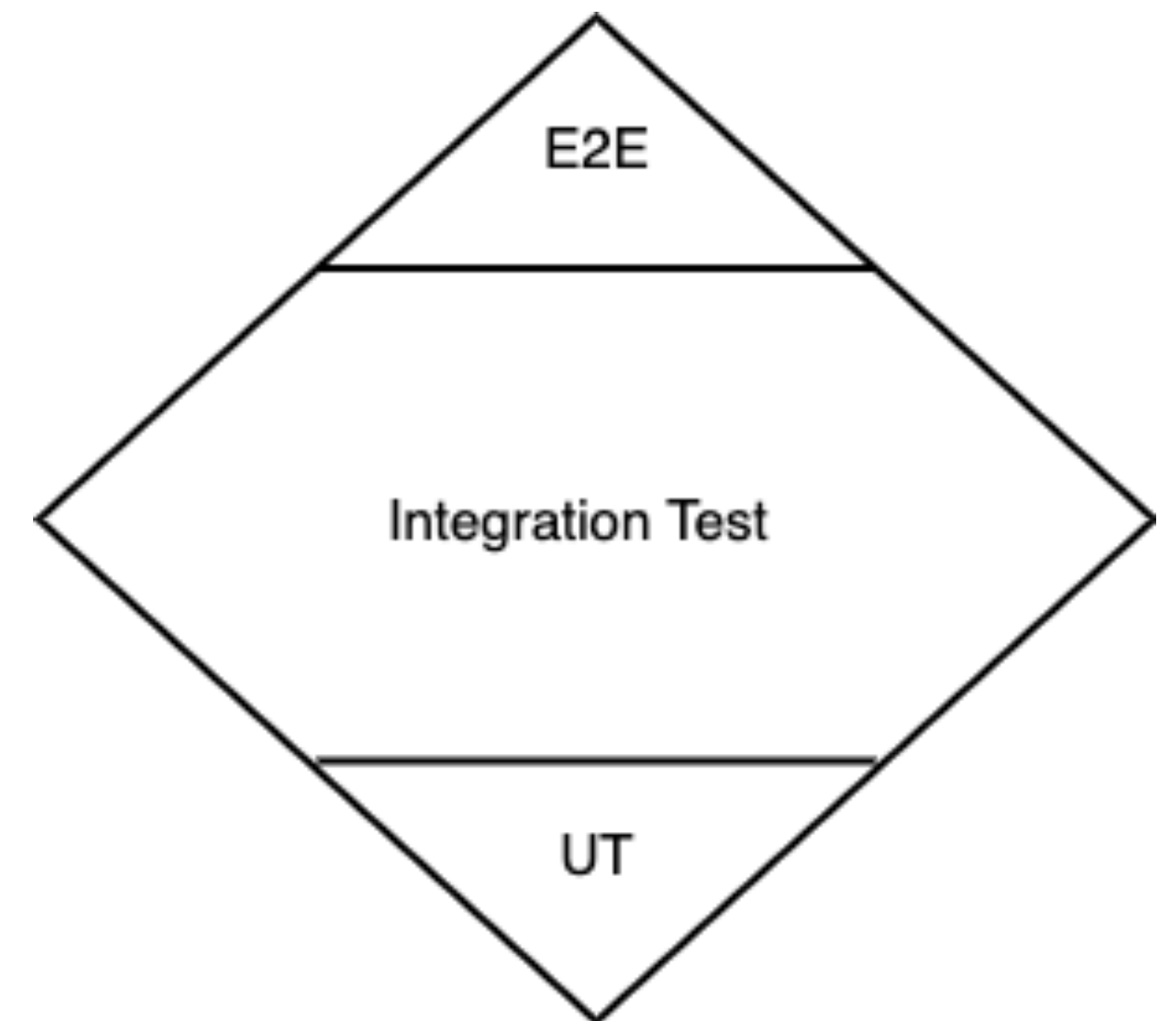
マイクロサービスアーキテクチャへの Integration Test 導入のすゝめ

~ソフトウェアテスト自動化カンファレンス2023~

(株)スマートショッピング
エンジニア 野島大誠

今日のお題

- 安心かつ高頻度デプロイ(=本番環境へのコード反映)を可能にすることを目指した結果、弊社では Integration Test の量が多いダイヤモンド型のテスト戦略となっています
- この背景、課題感を説明するとともに、運用での工夫点をお話しします
- 参考までにですが、弊社では現在1日3~4回程度デプロイを行っています



こんな悩みがある方のためになるかも

- マイクロサービスアーキテクチャのテスト戦略に関心がある方
- マイクロサービスアーキテクチャの複雑さに苦しんでいる方
- UT, E2E に物足りなさを感じ、Integration Test の導入に興味がある方

目次

- 抱えていた課題感😞
- Integration Test 運用での工夫💡
- まとめ👁️👁️

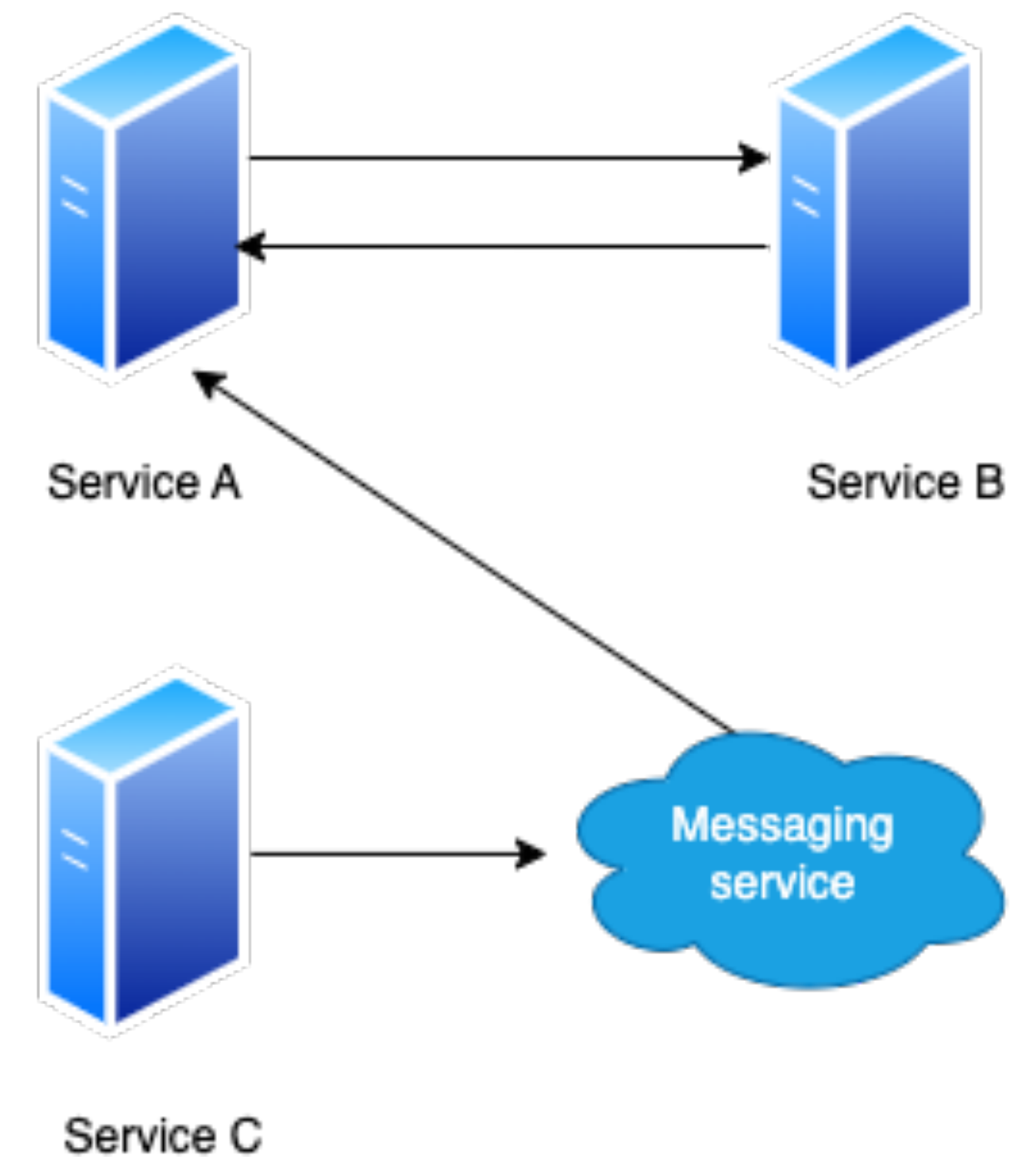
目次

- ▶ 抱えていた課題感😓
- Integration Test 運用での工夫💡
- まとめ👁👁

抱えていた課題感😞

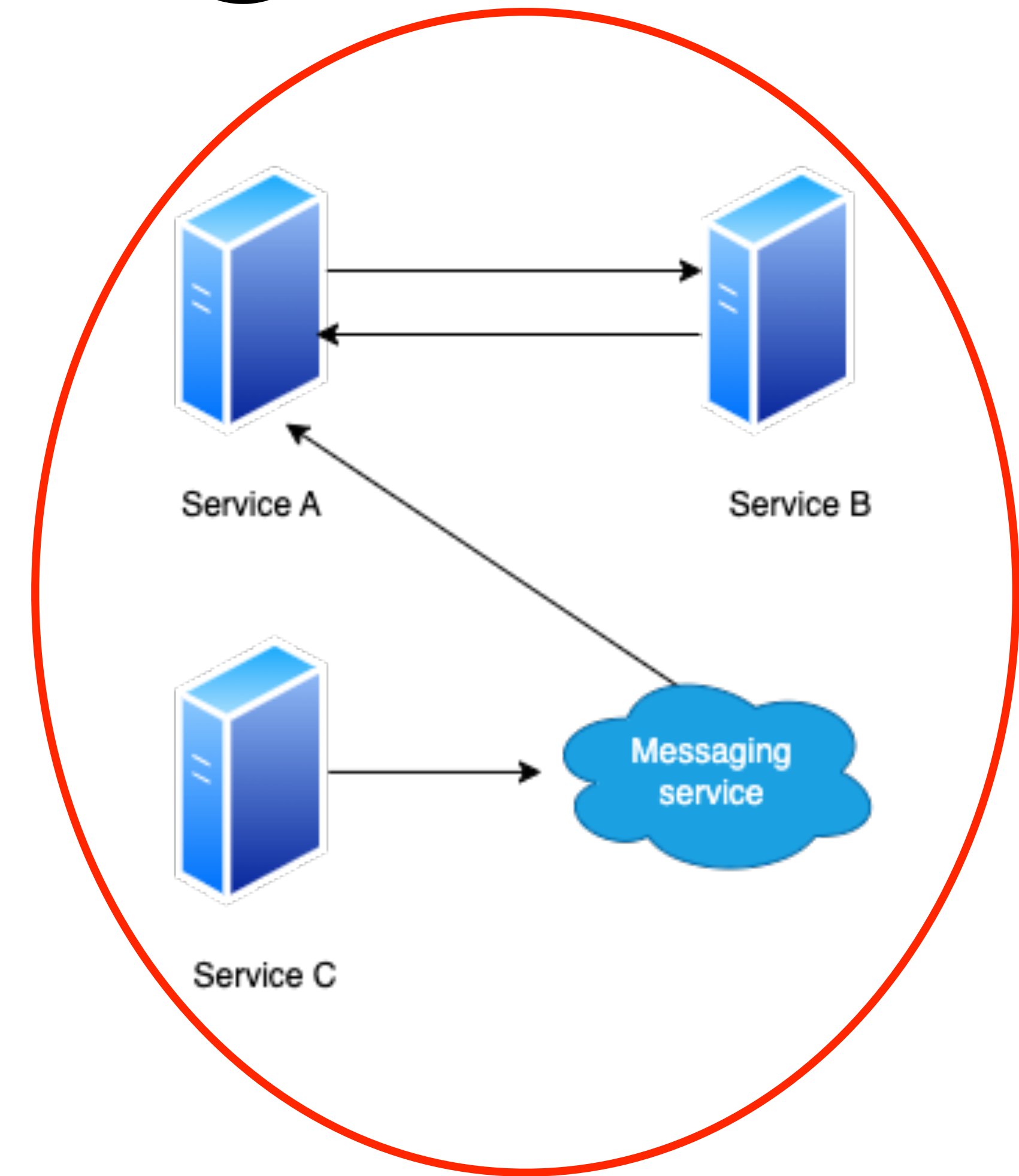
抱えていた課題感①

- マイクロサービスアーキテクチャ特有の難しさ



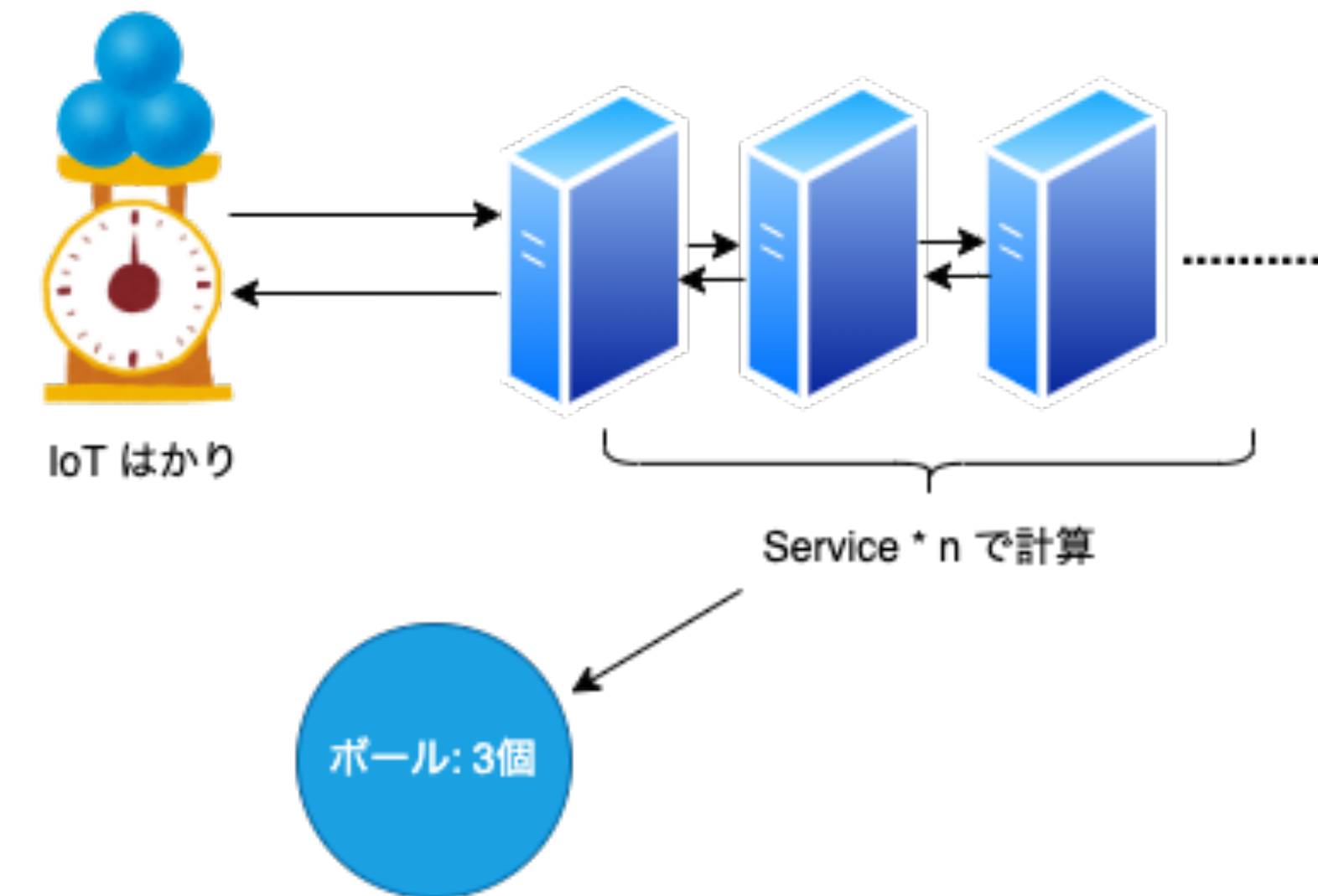
抱えていた課題感①

- マイクロサービスアーキテクチャ特有の難しさ
- ある計算結果を得る処理が複数のサービス、プロセスをまたがっておこなわれる



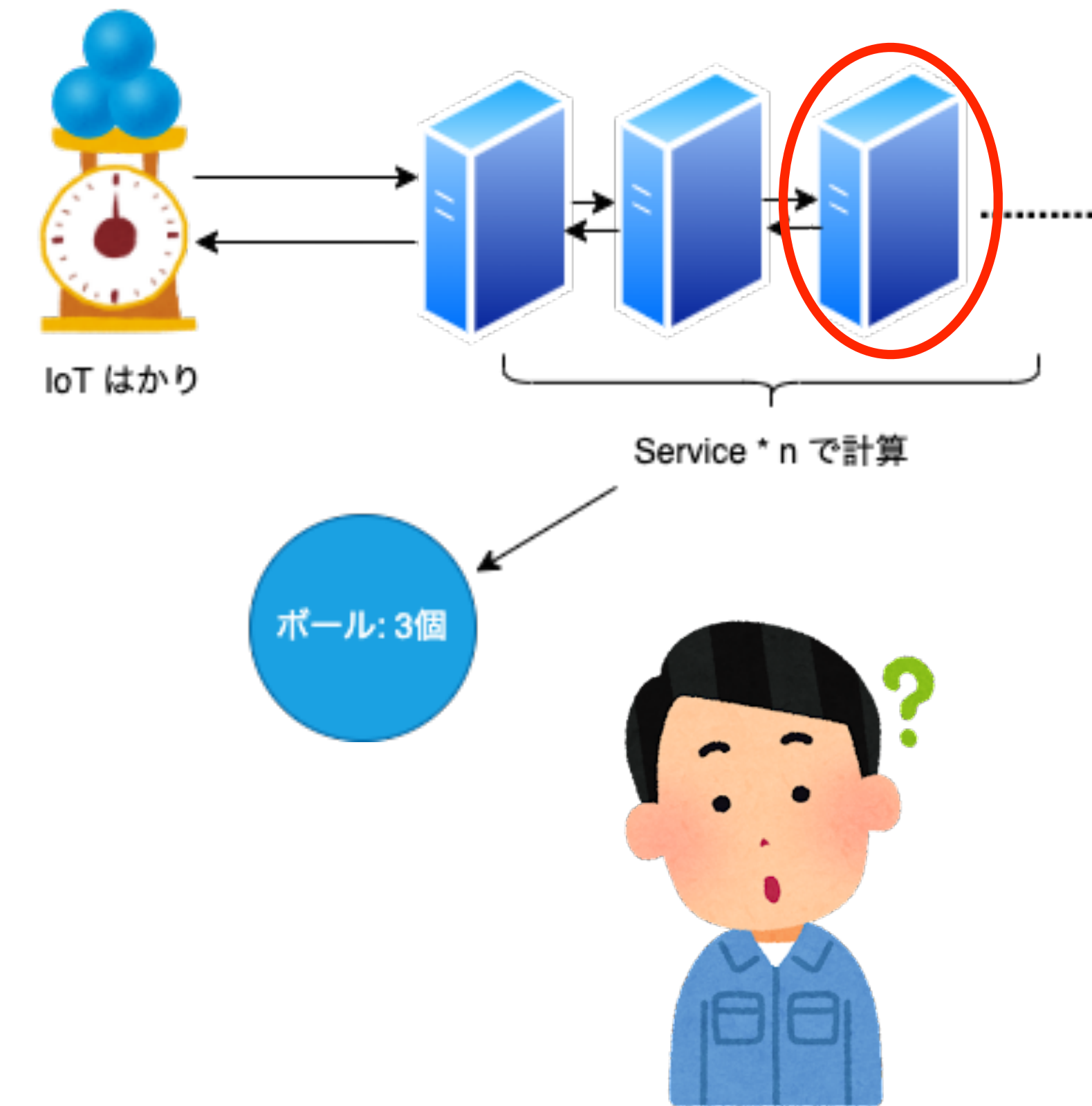
抱えていた課題感①

- マイクロサービスアーキテクチャ特有の難しさ
- ある計算結果を得る処理が複数のサービス、プロセスをまたがっておこなわれる
- **例: 弊社の在庫管理プロダクトは重量を測定して在庫数を推定する。この推定の結果を得るのに長く複雑な手続きを経る**



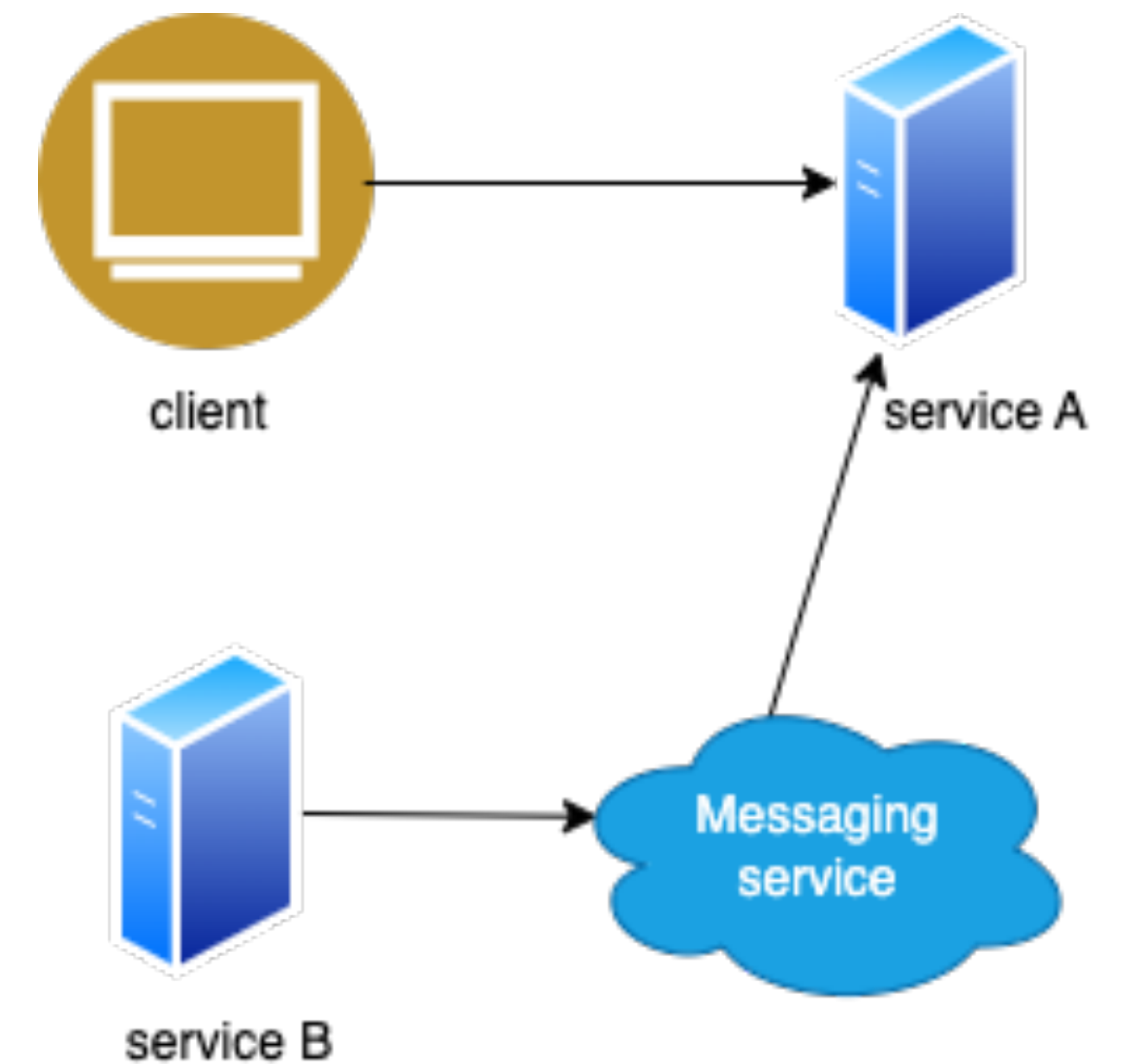
抱えていた課題感①

- マイクロサービスアーキテクチャ特有の難しさ
- ある計算結果を得る処理が、複数のサービス、プロセスをまたがっておこなわれる
- 例: 弊社の在庫管理プロダクトは重量を測定して在庫数を推定する。この推定の結果を得るのに長く複雑な手続きを経る
- **特定の計算のロジックにUTを書いても、在庫数の推定、という機能全体をテストで守れているか不安**😞



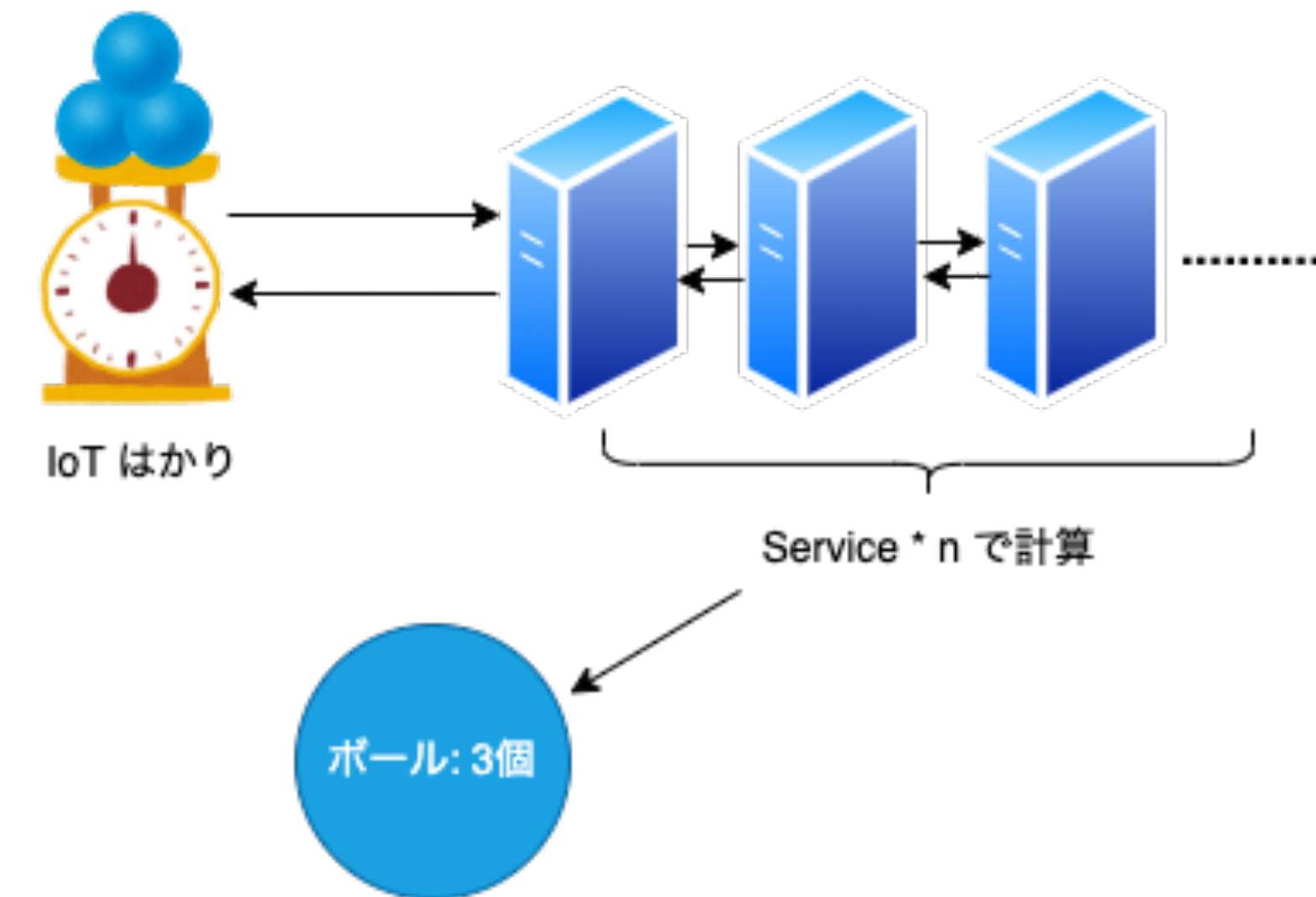
抱えていた課題感②

- 弊社プロダクト固有の課題で、画面契機のE2Eではカバーできない場所にコアな機能がある



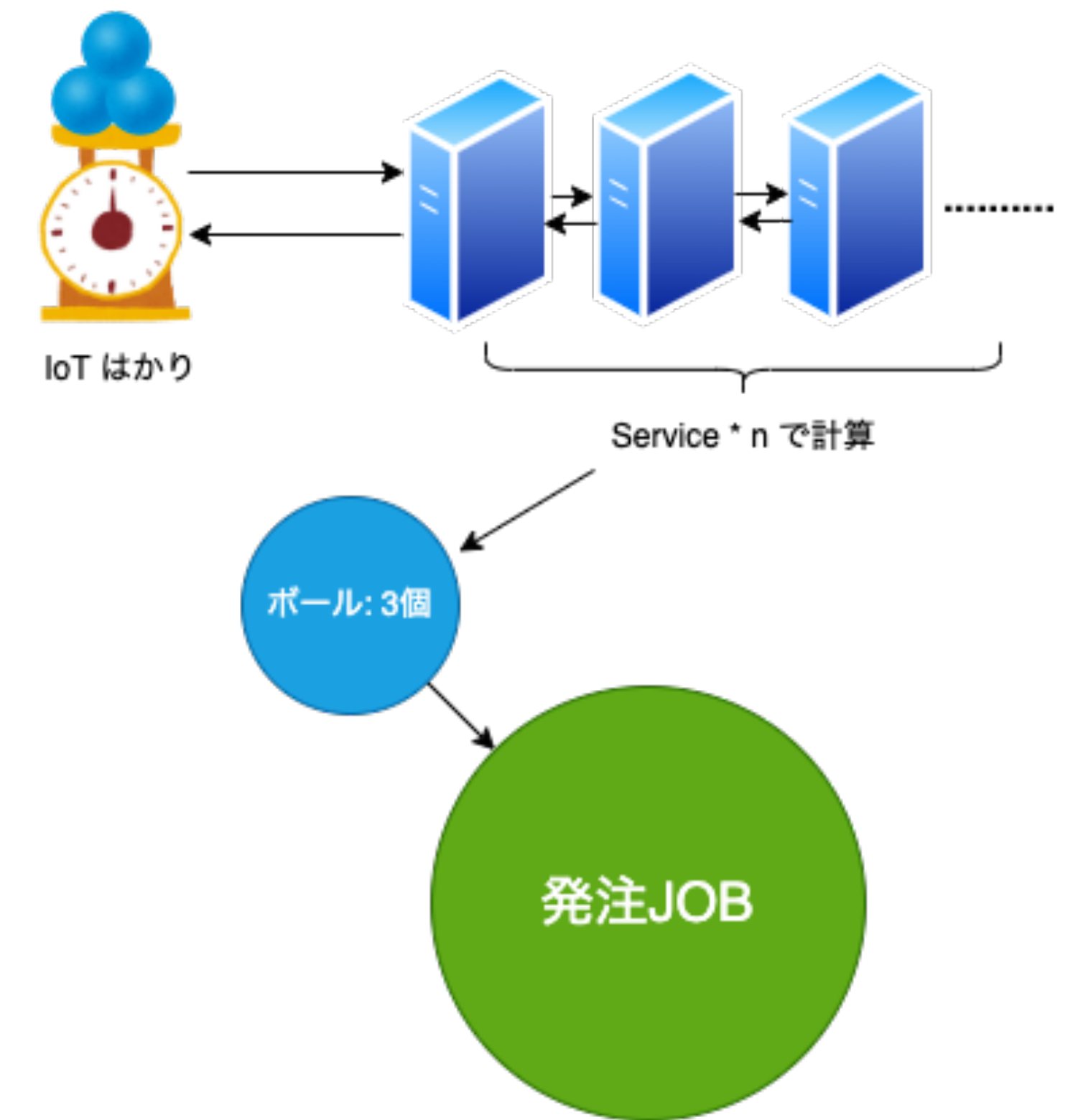
抱えていた課題感②

- 弊社プロダクト固有の課題間で、画面契機のE2Eではカバーできない場所にコアな機能がある
- **前掲の在庫推定はコアな機能であるが、IoTデバイス契機の処理**



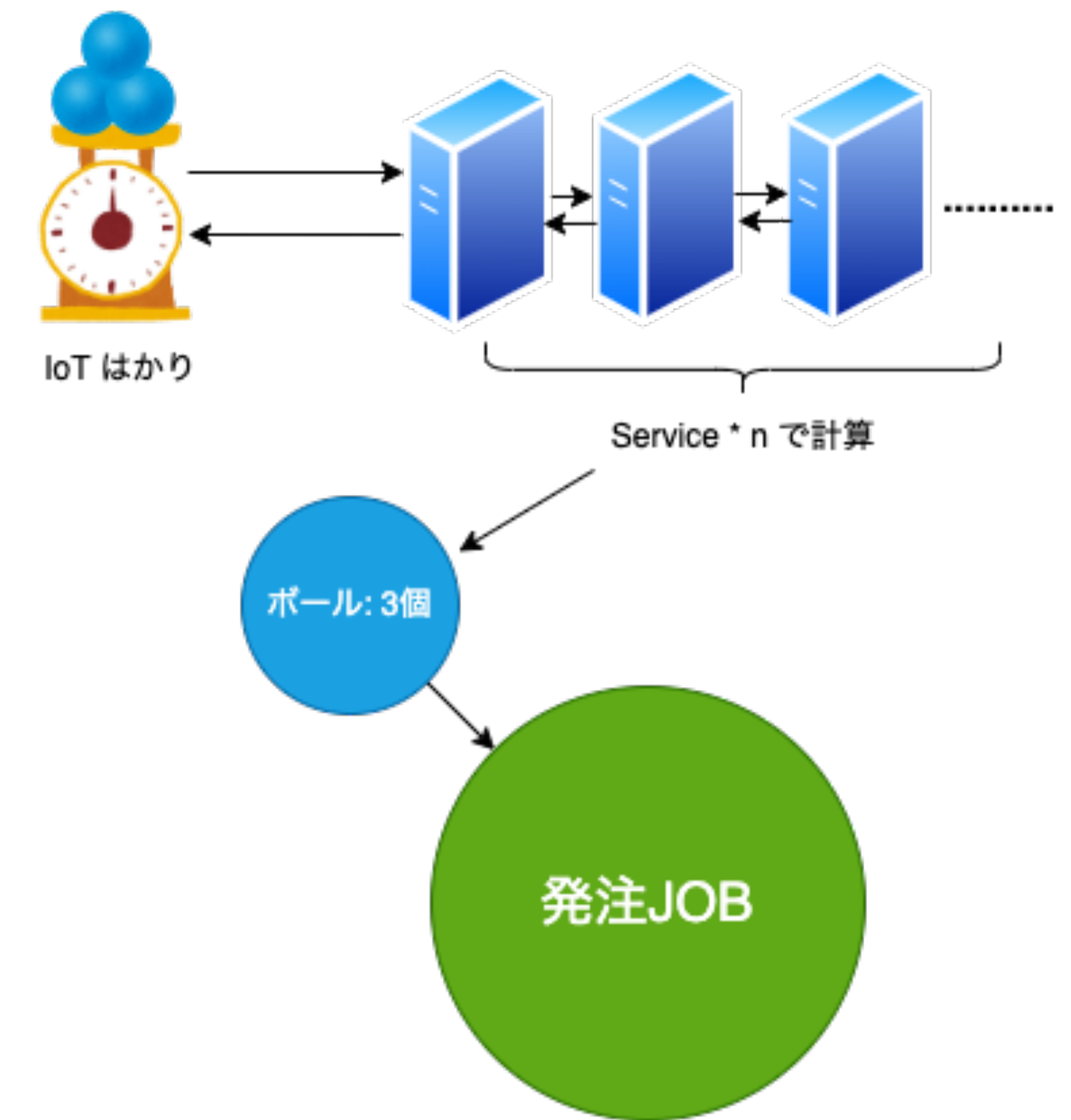
抱えていた課題感②

- 弊社プロダクト固有の課題間で、画面契機のE2Eではカバーできない場所にコアな機能がある
- 前掲の在庫推定はコア機能であるが、IoTデバイス契機の処理
- **また、その在庫推定の結果を受けた発注JOBというものがあり、これもコア機能**



抱えていた課題感②

- 弊社プロダクト固有の課題間で、画面契機のE2Eではカバーできない場所にコアな機能がある
- 前掲の在庫推定はコア機能であるが、IoTデバイス契機の処理
- また、その在庫推定の結果を受けた発注JOBというものがあり、これもコア機能
- **いずれも画面から発火する機能でない** 🙄



Integration Test の導入へ

- UT, E2Eを作成しても、カバーできないテスト範囲と拭えない不安感
- 不安感があるので、デプロイするのに二の足を踏むことがある
- これらの要素が重なり、Integration Test を導入

目次

- 抱えていた課題感😞
- ▶ **Integration Test 運用での工夫💡**
- まとめ👁👁

Integration Test 運用での工夫💡

Integration Test 運用での工夫①

- テストケース選定における工夫



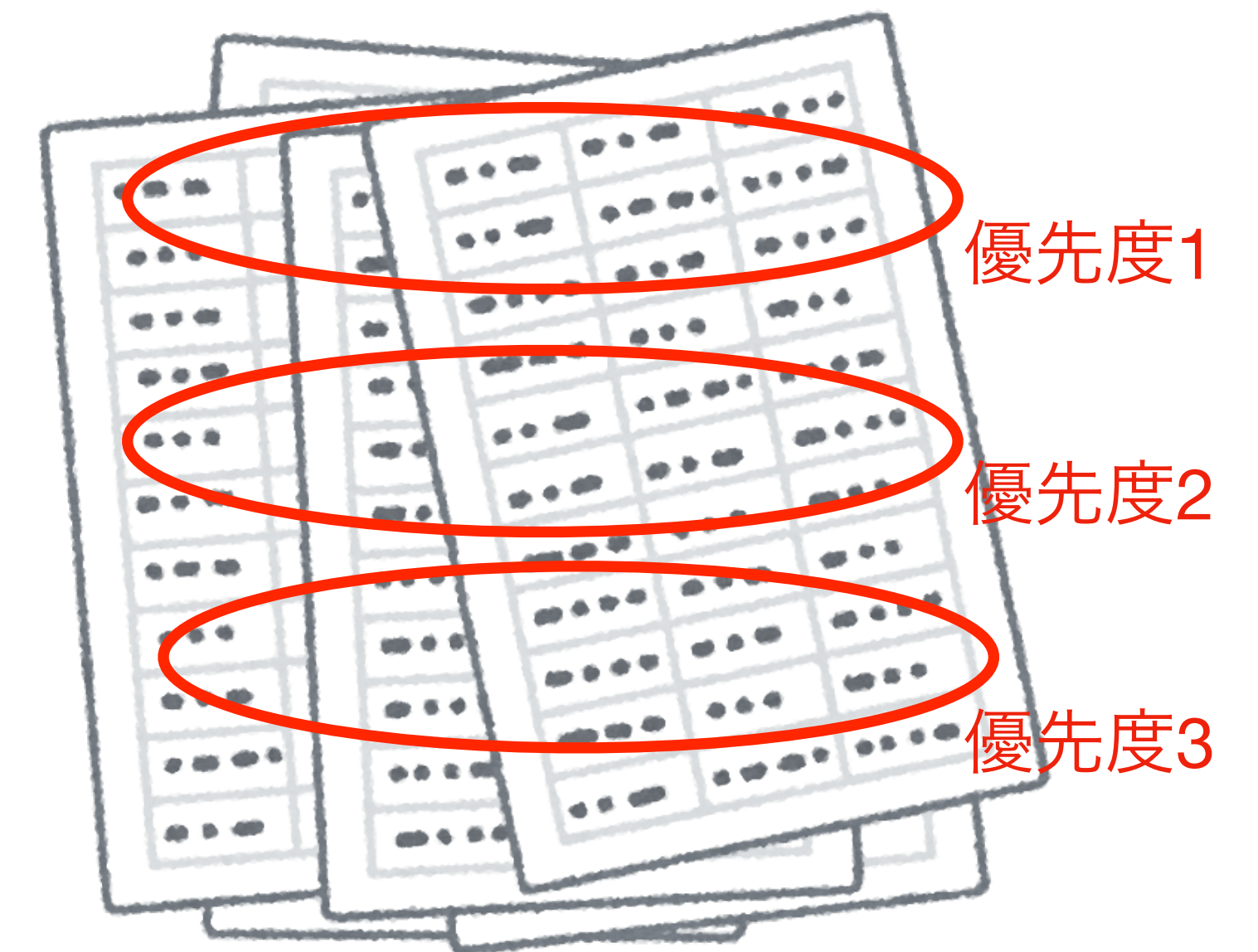
Integration Test 運用での工夫①

- テストケース選定における工夫
- 弊社はベンチャー企業であり、機能開発により工数をかけたいので、テスト実装にかけられる時間に制限がある



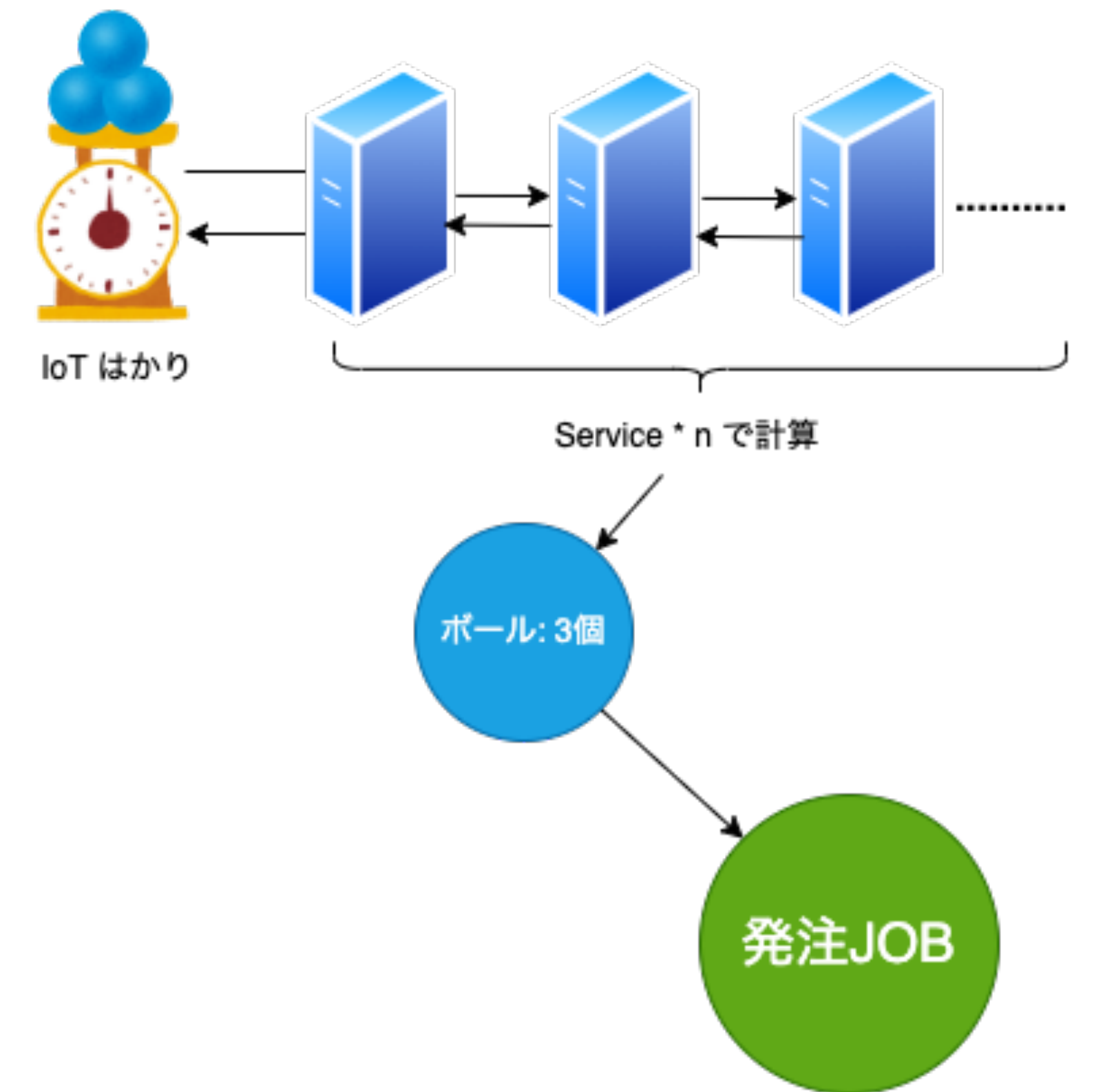
Integration Test 運用での工夫①

- テストケース選定における工夫
- 弊社はベンチャー企業であり、機能開発により工数をかけたいので、テスト実装にかけられる時間に制限がある
- **そこで重要な機能と、考えられるテストケースをリストアップし、優先度をつけてテストの実装をすすめた**



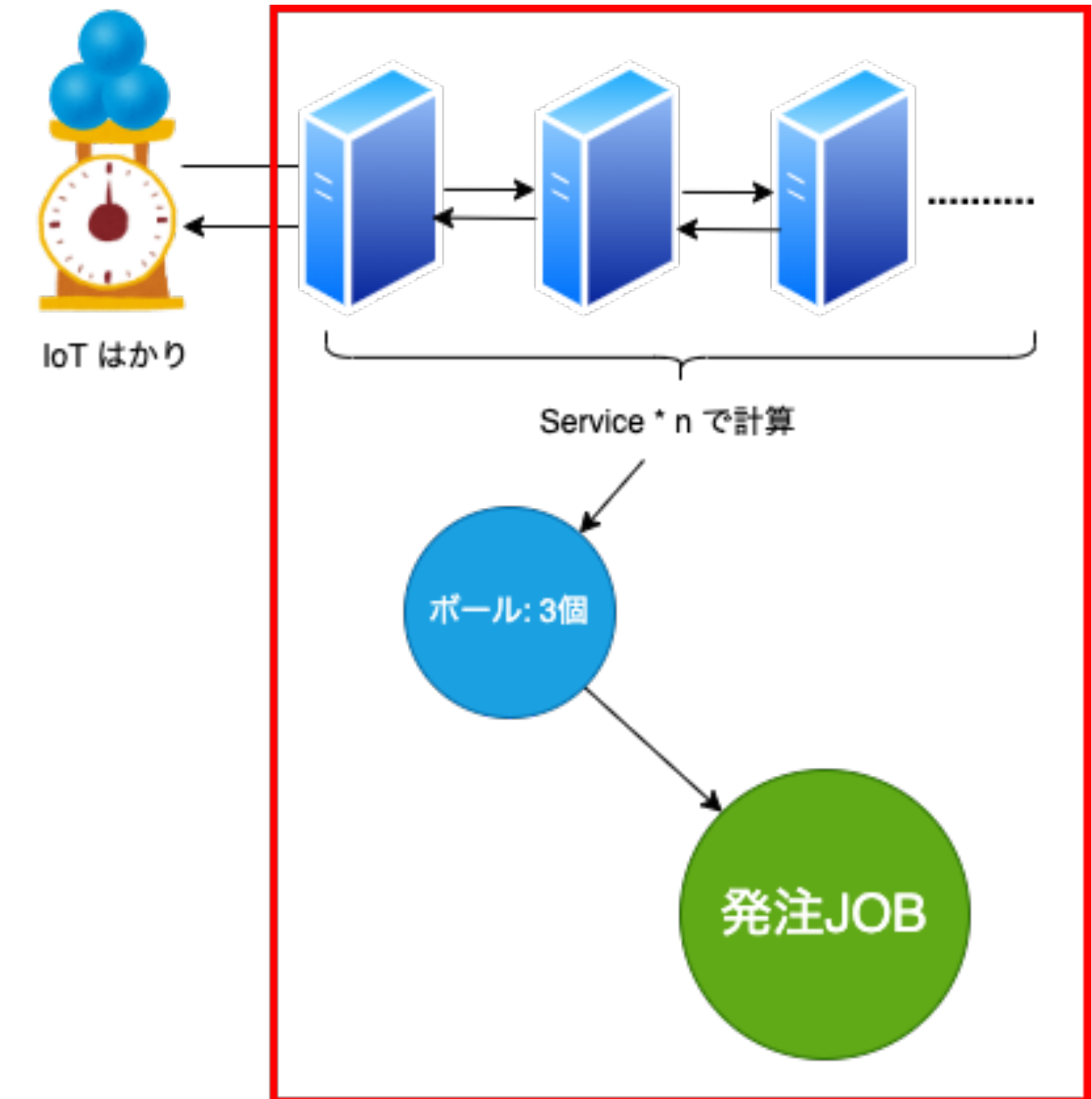
Integration Test 運用での工夫②

- テスト範囲設定における工夫



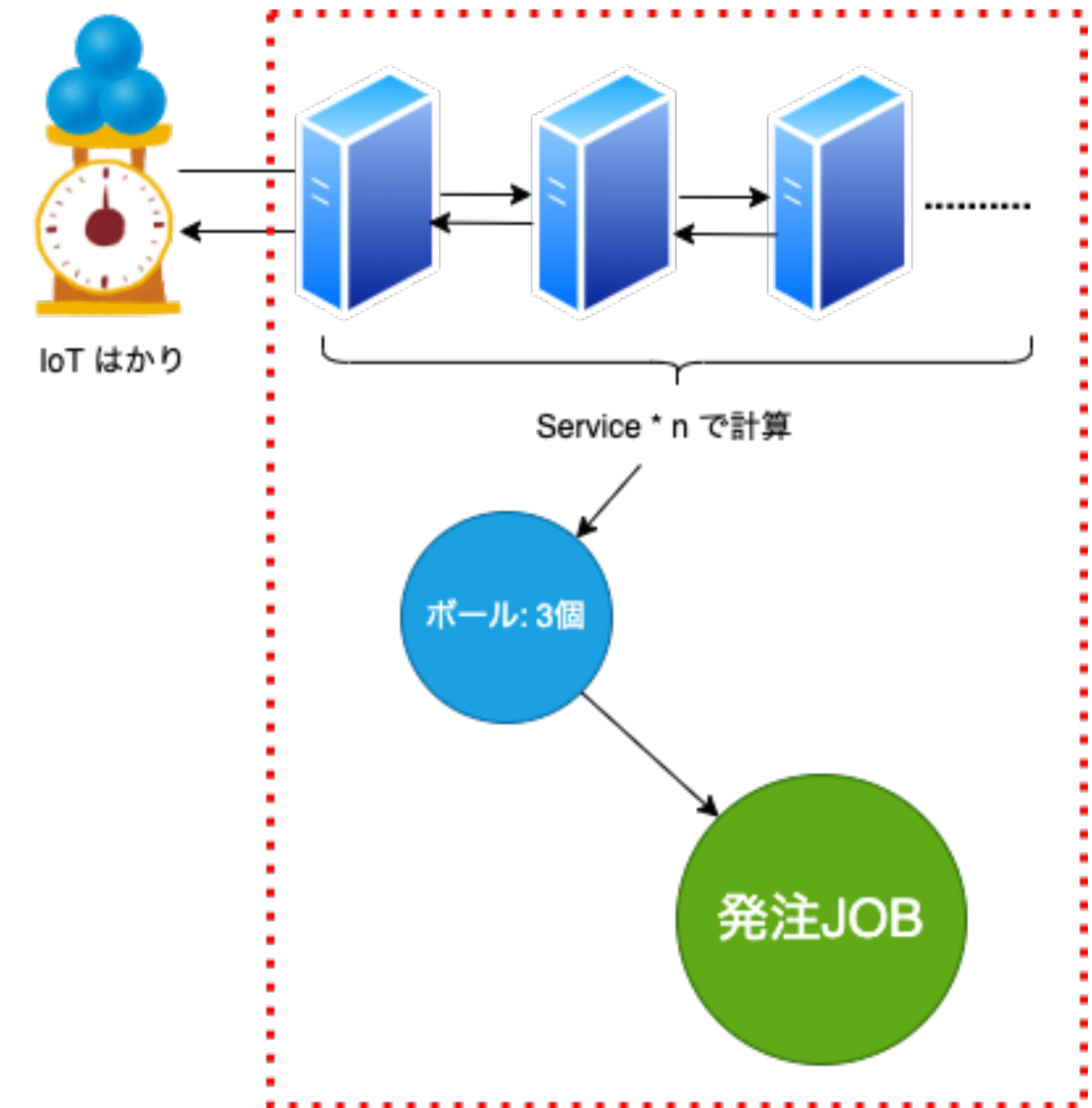
Integration Test 運用での工夫②

- テスト範囲設定における工夫
- アプリ間の協調動作をテストしたいが、テスト範囲を大きくし過ぎるとFlakyになる



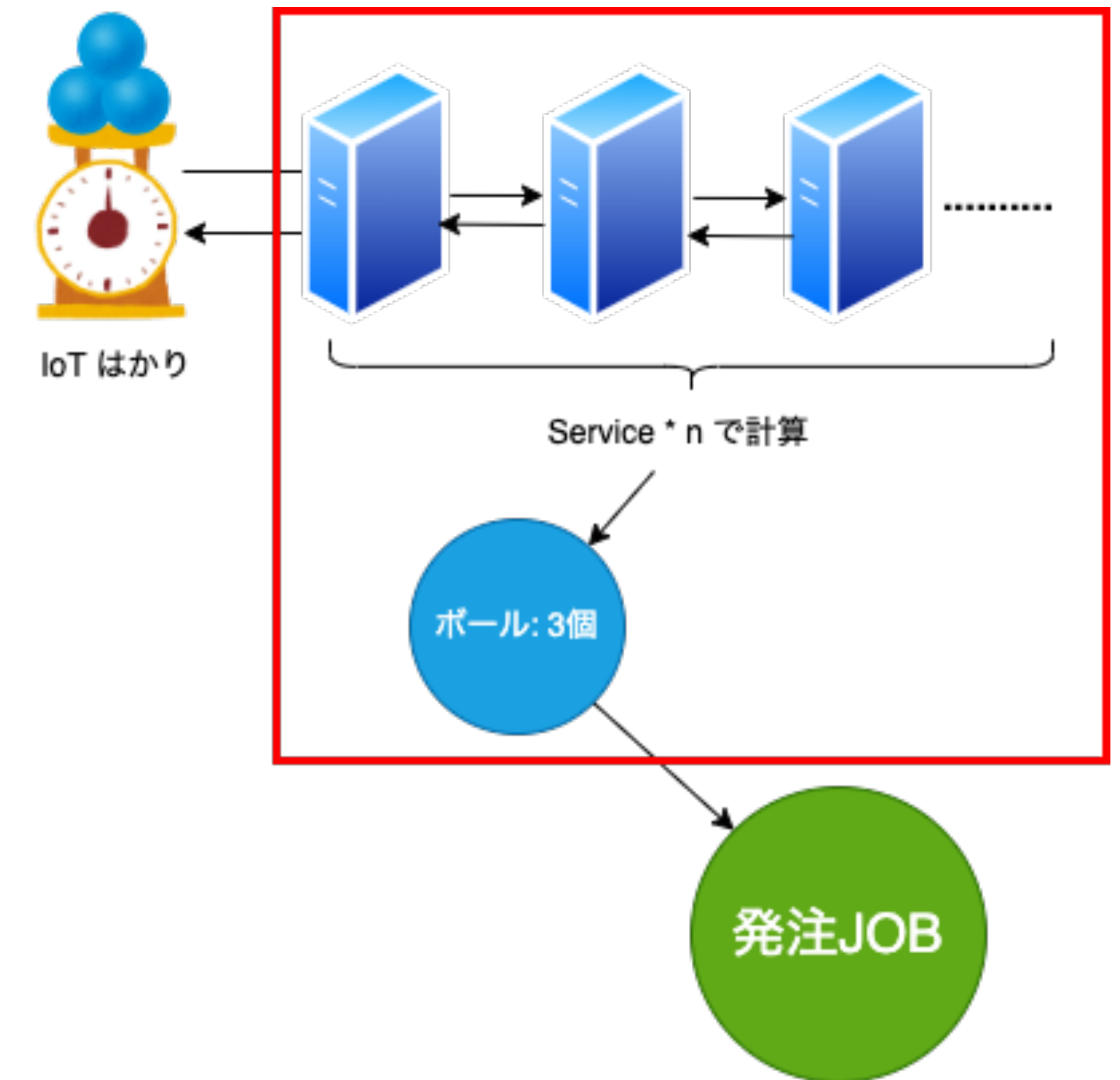
Integration Test 運用での工夫②

- テスト範囲設定における工夫
- アプリ間の協調動作をテストしたいが、テスト範囲を大きくし過ぎるとFlakyになる
- **テスト範囲を柔軟に設定し、それらを総合的にみて重要な箇所をテストでカバーできていることを保証**



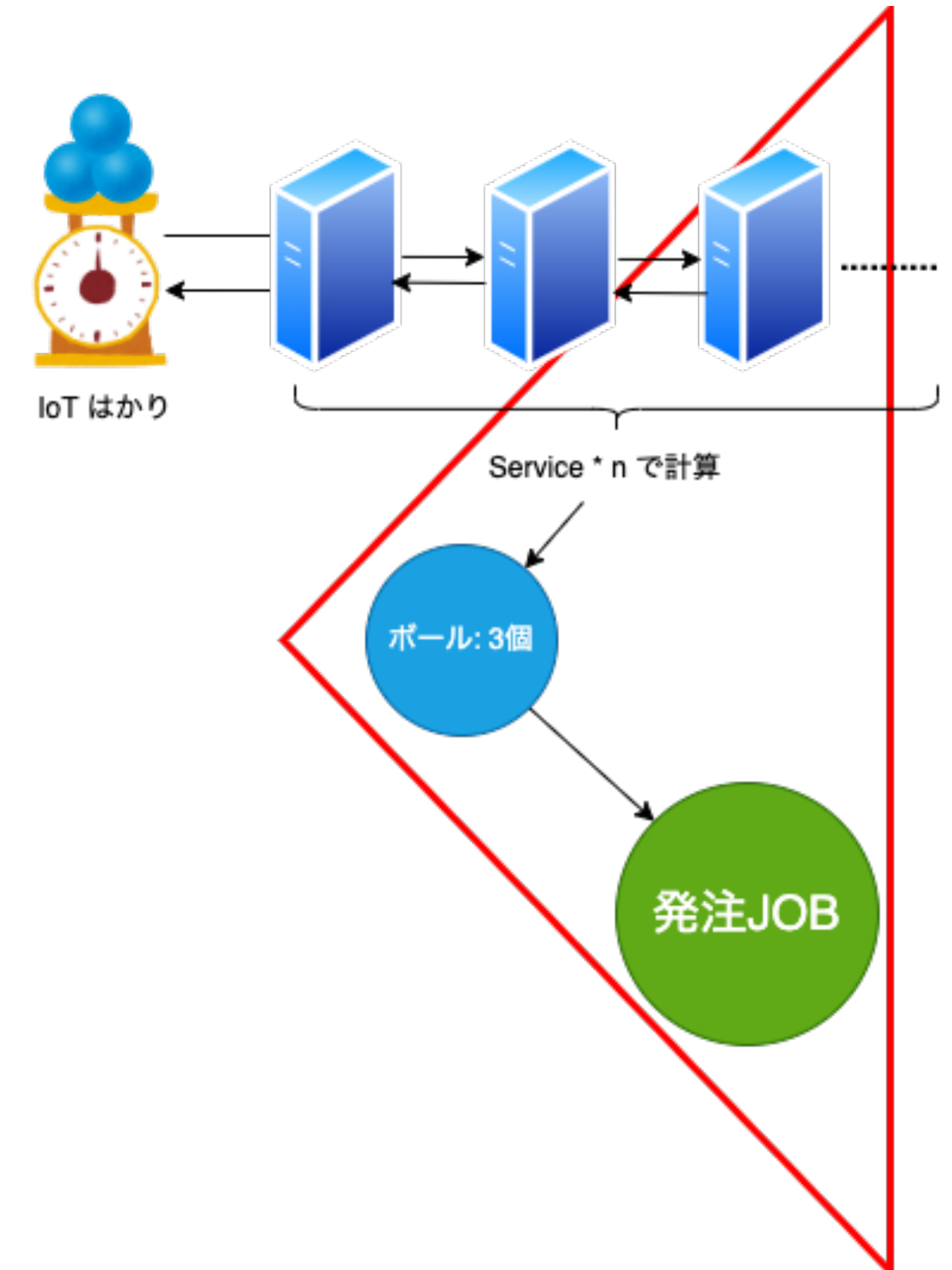
Integration Test 運用での工夫②

- テスト範囲設定における工夫
- アプリ間の協調動作をテストしたいが、テスト範囲を大きくし過ぎるとFlakyになる
- テスト範囲を柔軟に設定し、それらを総合的にみて重要な箇所をテストでカバーできていることを保証
 - 例) **在庫推定範囲** * 発注範囲



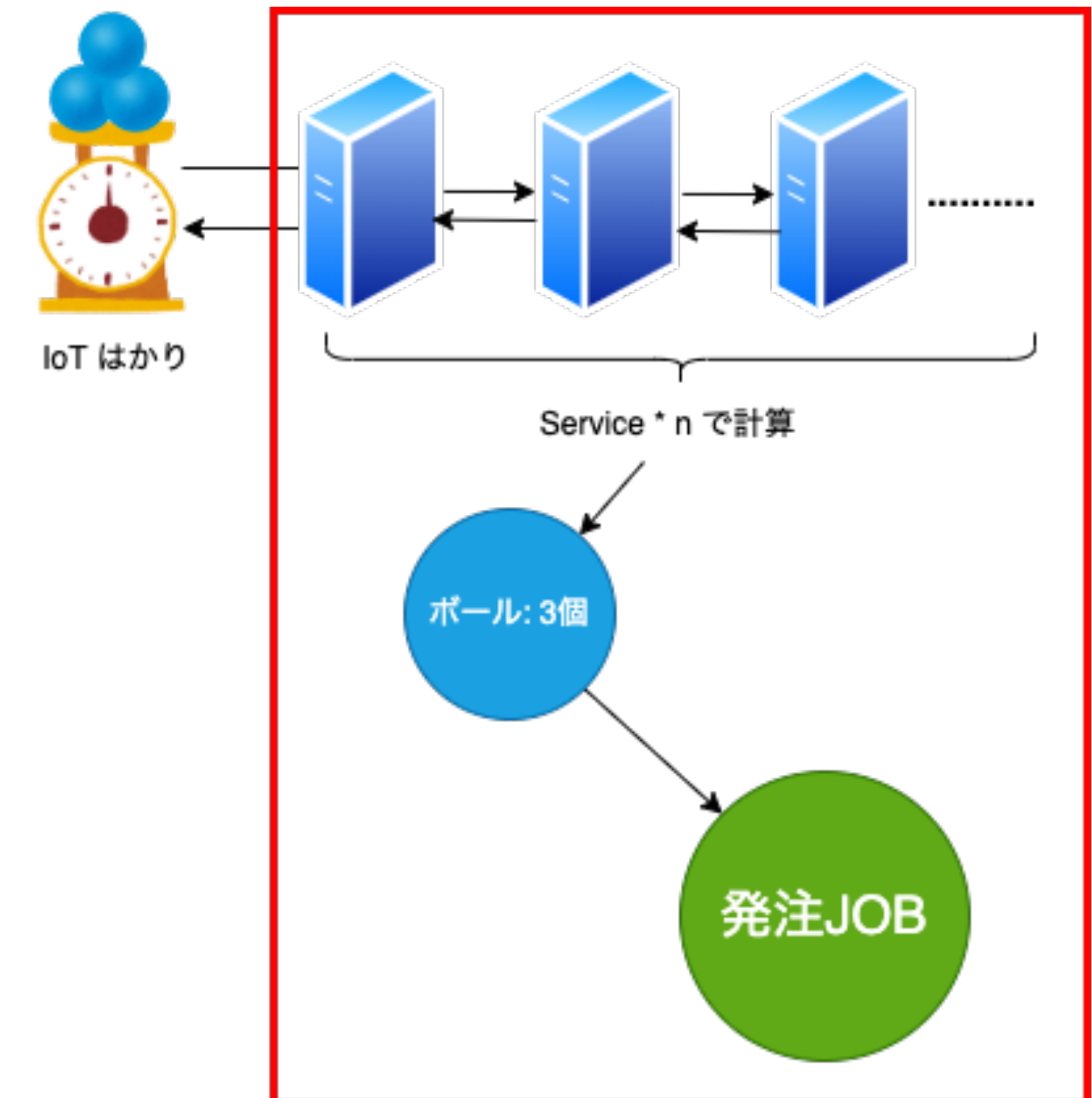
Integration Test 運用での工夫②

- テスト範囲設定における工夫
- アプリ間の協調動作をテストしたいが、テスト範囲を大きくし過ぎるとFlakyになる
- テスト範囲を柔軟に設定し、それらを総合的にみて重要な箇所をテストでカバーできていることを保証
 - 例) 在庫推定範囲 * 発注範囲



Integration Test 運用での工夫②

- テスト範囲設定における工夫
- アプリ間の協調動作をテストしたいが、テスト範囲を大きくし過ぎるとFlakyになる
- テスト範囲を柔軟に設定し、それらを総合的にみて重要な箇所をテストでカバーできていることを保証
 - 例) **在庫推定範囲 * 発注範囲**



目次

- 抱えていた課題感😓
- Integration Test 運用での工夫💡
- ▶ まとめ👁👁

まとめ👁👁

まとめ

- Integration Test を導入するに至った課題感と、運用上の工夫を共有しました
 - UT, E2Eでカバーできない範囲にコアな機能がある
 - テストケースの優先度づけと柔軟なテスト範囲の設定
- 結果、Integration Test が厚めのダイヤモンド型のテスト戦略となっています
- 組織ごとに適したテスト戦略があると思いますが、弊社の事例がテスト戦略を考える際の一助となれば幸いです

