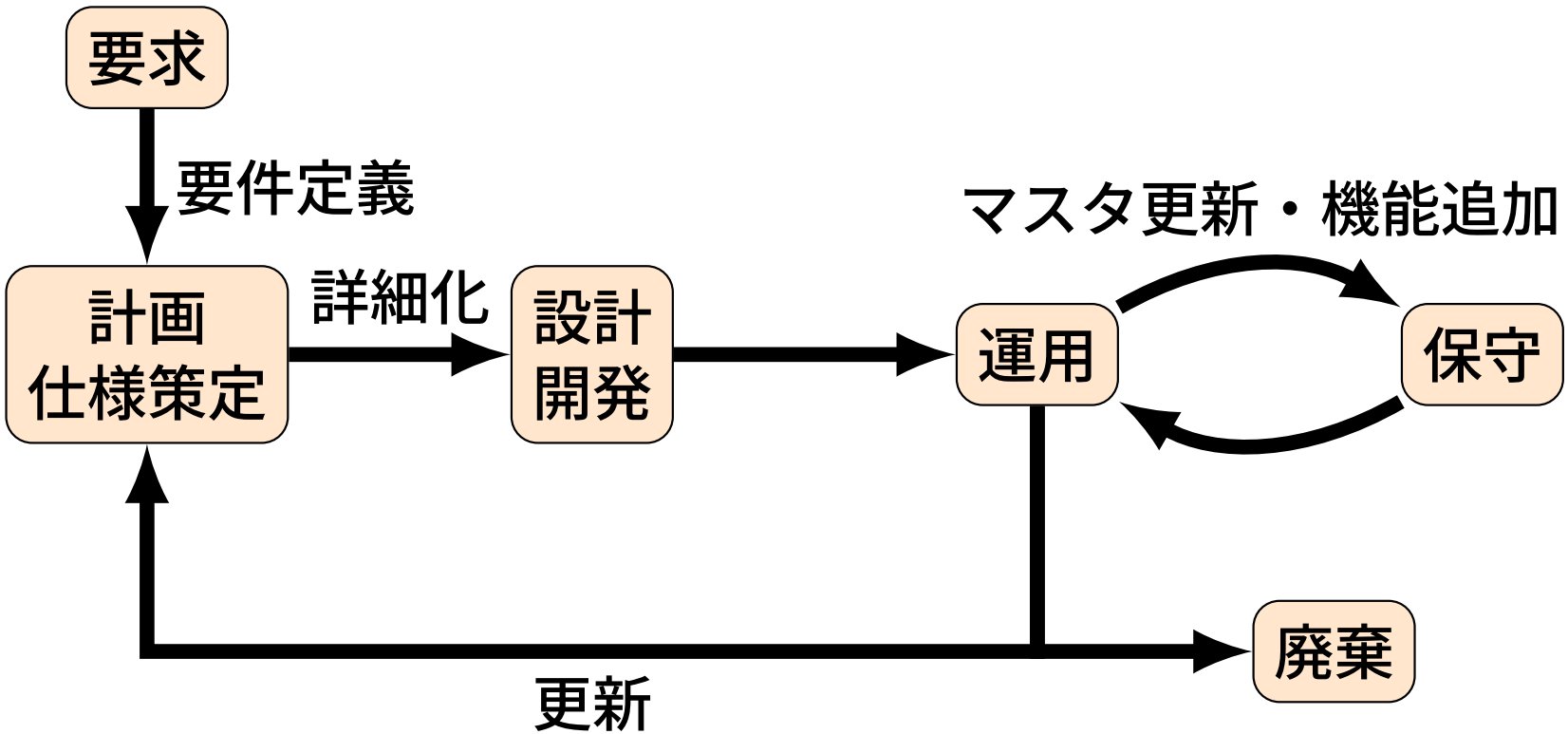
システムのライフサイクル

情報科学の世界 2
2023 年度前期
佐賀大学理工学部 只木進一

システムのライフサイクル



システムの計画

- 定性的・非技術的要求を定量的・技術的要件へ
  - 業務を行う部署と情報関連部署の連携が必須
  - 業務改革が連動しなければ却って非効率になる危険性
- 他のシステムとの連携も意識
- 実現可能性・費用対効果・運用コスト・利用可能性も検討
  - 運用できなければ意味がない
  - 使ってもらえなければ意味がない

要件定義: 業務を行う部署から

- システムの目的を整理
  - どの業務をシステム化するか
  - システム化で何を可能とするのか
- 業務フローの明確化
  - 何をどのような順序で行っているか
  - ここが難関
- 業務改革は必須
  - 紙で行っていた時との違い
  - システム化で省略できること
  - システム化で新たにできること
  - 法律の制約

機能要件と非機能要件

- 機能要件
  - 業務に関する機能
  - 「○○ができること」
  - どのようなデータを保持するか
- 非機能要件
  - 性能: 同時アクセス○人
  - 保守: 「障害時に 1 時間以内に対応」
  - セキュリティ
  - 旧システムからの移行
  - 利用終了時の対応: 機材撤去、後継システムへのデータ移行等
  - 開発プロジェクトの体制

詳細仕様

- 要件と技術仕様との調整
  - 要件を技術の用語に翻訳
  - 業務部門とシステム開発者との調整
- 画面イメージ (mockup)
- 処理の流れ
  - 画面遷移
  - データ更新の時刻
- データベース設計
- 非常に重要な過程。失敗すると大きな遅れと費用が発生

質問

要件定義を行うのは、業務を担当する部署です。その際に、対象業務のワークフローの明確化が重要です。しかし、それは簡単なことではありません。その理由を考えましょう。

システムの開発

- 外部設計
  - 機能要件
  - ユーザインターフェース
  - 他システムとの連携
- 内部設計
  - システムのモジュール化
  - インターフェイス詳細設計
- プログラム設計
  - データ構造
  - プログラムモジュール
  - モジュール間連携
- メタ言語の活用

モジュール化

- システムをできるだけ独立した部品に分割する
  - 業務毎の機能
  - 共通的機能
- MVC (Model-View-Control)
  - 業務のモデル
  - ユーザインターフェイス
  - 作業間の遷移
- モジュール化の利点
  - モジュール毎に機能・性能を評価できる
  - 変更や修正範囲を限定できる
  - インターフェースの調整は必要

開発の後半

- プログラミング
- テスト
  - 単体テスト: 各部品のテスト
  - 結合テスト: 部品を連結した後のテスト
- 検収
  - 機能性能要件を満たしているか

品質保証

- ソフトウェア全般の質を保証
- コーディング規約
  - 誰が書いても同じ品質
- テスト体制
  - 開発者とは違う人がテスト
- 開発とは独立した品質保証部門

質問

システム開発プロジェクトでは、品質管理が重要な要素です。品質管理部署が開発部署と分かれているべきである理由を考えましょう。

システムの日常的管理

- システム構成の現状把握
  - ハードウェア、ソフトウェア、ネットワーク、マニュアル
- 利用状況、性能状況
  - ディスク、メモリ
- セキュリティ状況
  - 脆弱性情報
  - 不正侵入の企て

利用者管理・資源管理・運用管理

- 利用者管理
  - 利用者登録・削除
  - 権限付与
- 資源管理
  - ファイル領域
  - メモリ、CPU 割り当て
- 運用管理
  - 運用規則との整合性
  - 人的資源配置

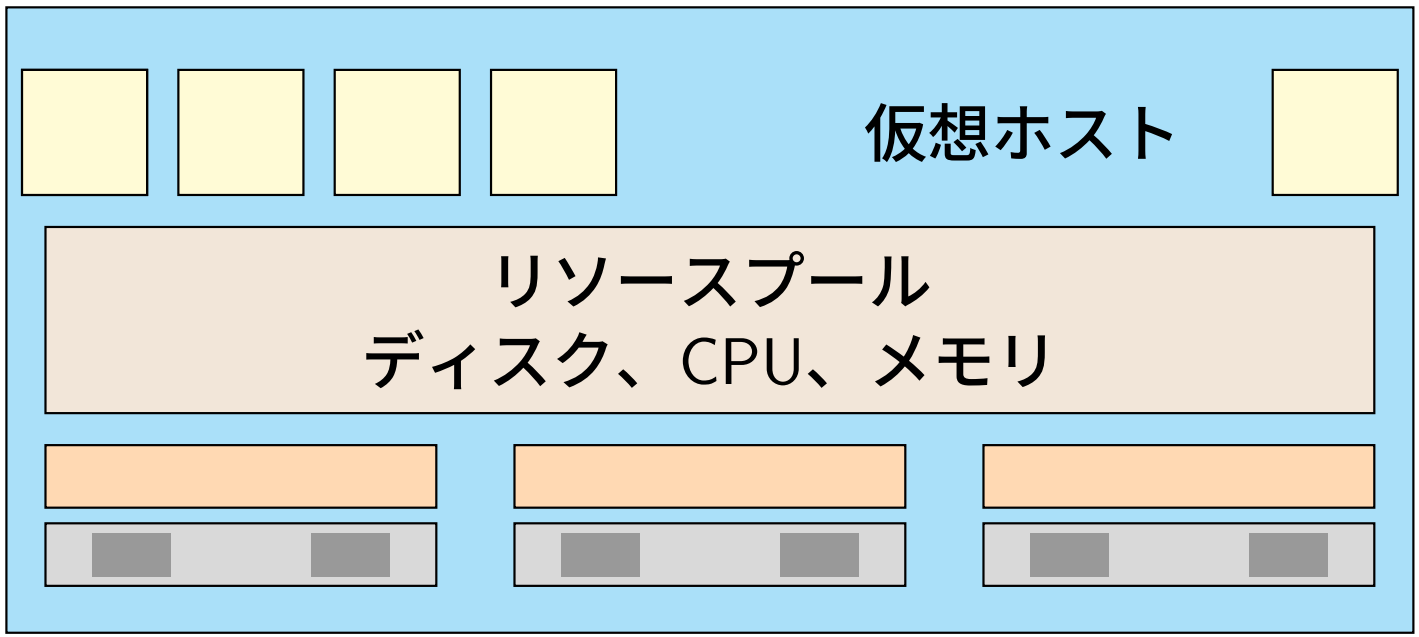
システムの障害と対策

- ソフトウェア障害
  - 仕様との不整合
  - 追加・変更が必要となる場合
- セキュリティ脆弱性
  - パッチ、設定見直し

ハードウェア障害

- ディスク
- ネットワーク
- 電源
- 性能不足
- 冗長化・多重化で対策

サーバ仮想化



- ハードウェアとサーバとの関係を柔軟に構成する
  - 複数のハードウェアにまたがってリソースプールを作ることで冗長化を実現
  - リソースプール上に多数の仮想ホストを実装


AC
AC
NTTファイナンス

質問

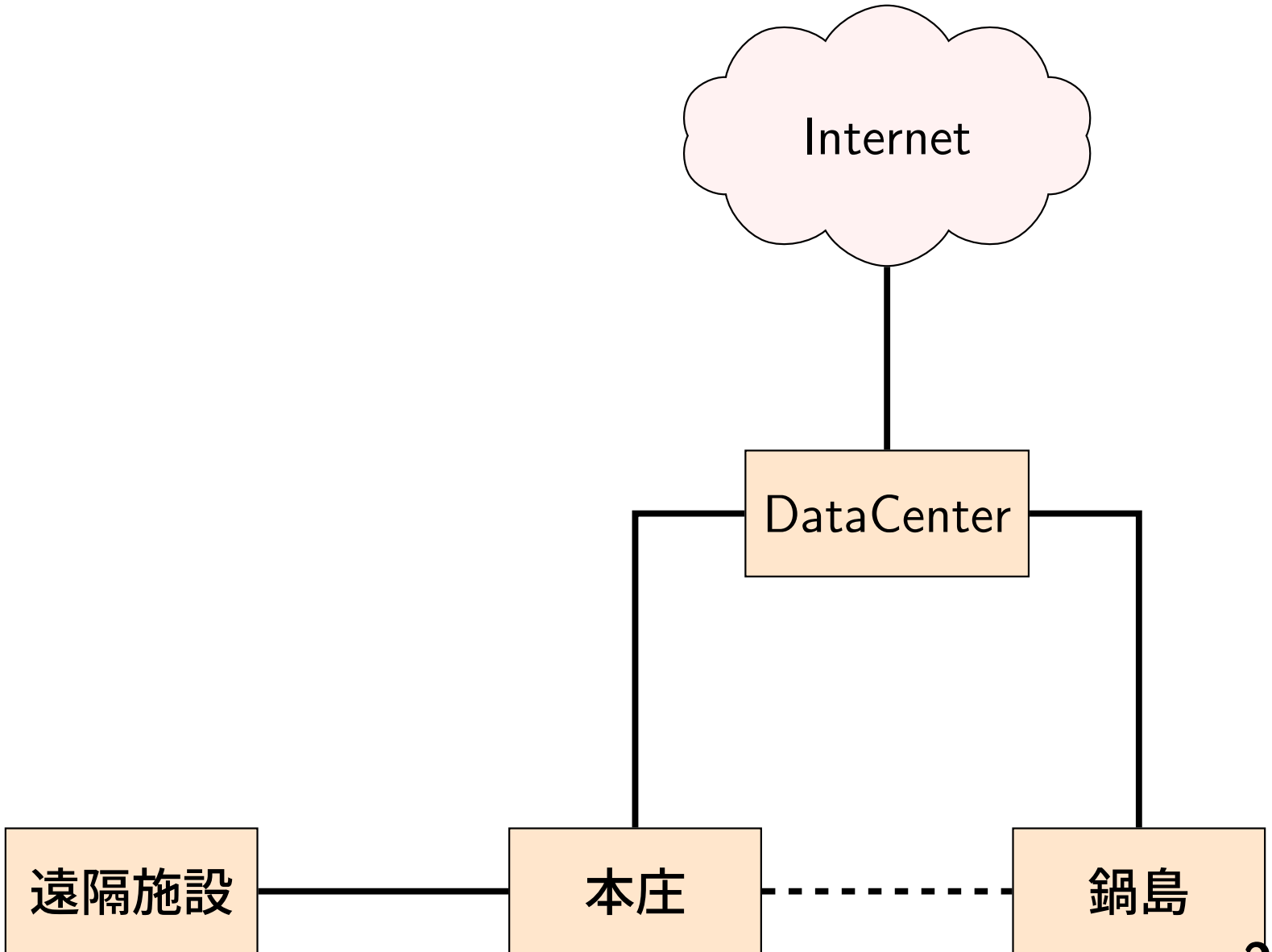
サーバ仮想化は、単にサーバ機を購入して構築することに比べて、どのようなメリットがあるのでしょうか。

冗長化・多重化技術: ディスクと電源

ディスク冗長化・多重化 (RAID、ホットスワップ)

電源多重化

冗長化・多重化技術: ネットワーク



利用者管理

- **端末認証**
- Active Directory : Windows
- LDAP : UNIX, Linux, Mac
- Web **アプリケーション**
- SSO (Single Sign-On)
- Shibboleth
- OpenAM

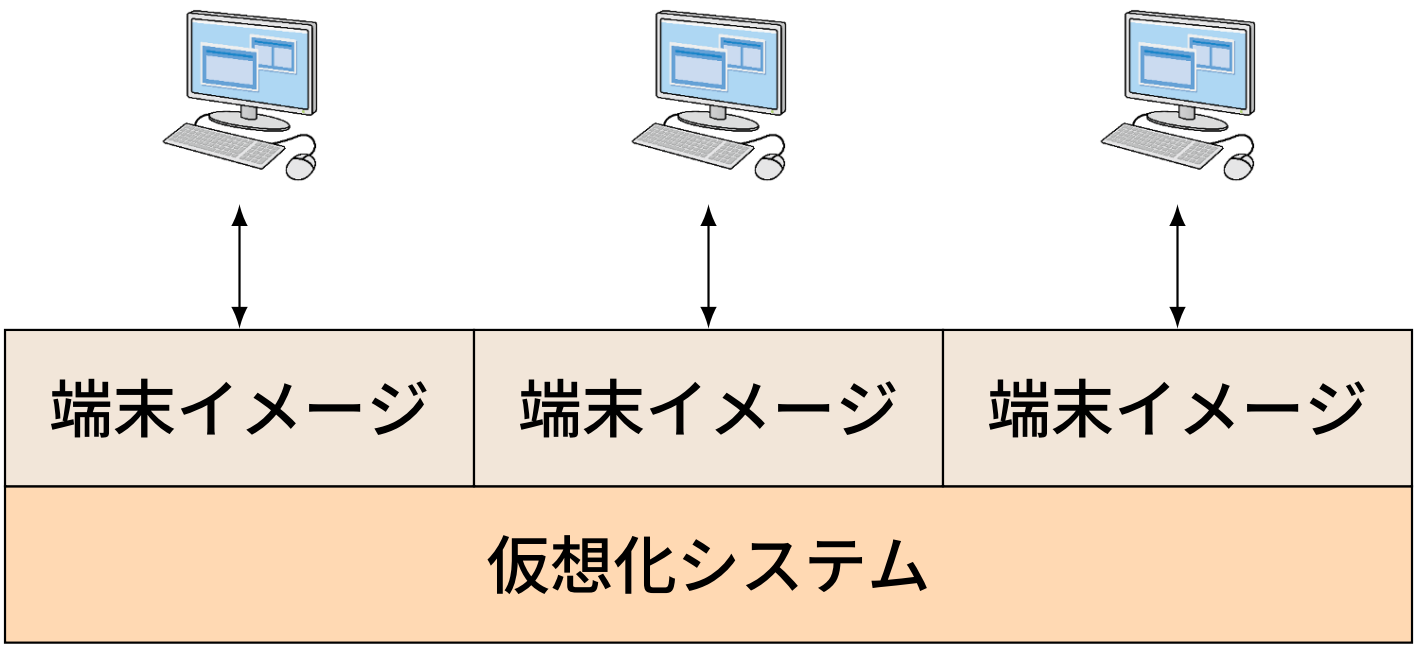
端末管理: ネットワークマウント

- 利用者のファイルを端末に持たない
- 利用者ファイルはファイルサーバに
- ネットワークを介して利用

端末管理:Thin Client

- 端末にハードディスクを持たない
- Network Boot 方式
- 起動イメージをネットワークを介して取得
- 必要に応じて、ファイルサーバをマウント

端末管理: 画面転送方式



運用の観点から見たクラウド

- SaaS : Software as a Service
  - ソフトウェアを借りる
- PaaS: Platform as a Service
  - OS やミドルウェアを借りる
- IaaS: Infrastructure as a Service
  - 仮想サーバを借りる
- DaaS: Desktop as a Service

SaaS 利用

- 構築、運用コストの大幅削減
- 自由度は低くなる
- 例: O365
  - メール
  - スケジュール
  - ファイル共有
  - sharepoint サーバ

PaaS 利用

- OS やミドルウェア更新コスト削減
- アプリケーションサービスに注力できる
- 例: さくらレンタルサーバ
  - Web サーバ + CMS
- 例: AWS

IaaS 利用

- ハードウェア保有コスト削減
- ハードウェア故障対策、電源、空調のコスト削減
- 例: さくら VPS
  - CPU、メモリ、ディスクを指定して借用
  - 自分で OS からインストール
- 例: Microsoft Azure

クラウドサービスの利点

- サーバのハードウェア、保管場所、電源、空調等の費用削減
- 短時間でのサービス開始
- ソフトウェア更新を任せる
- セキュリティ対策の外部化

データセンター

- 強固な建物、電源、空調、入退室管理
- 電源やネットワーク回線を冗長化して災害対策
- 死活監視等のサービス
- 利用のメリット
  - 機器を自組織に置かず、専用の施設に預ける
  - 電源、空調、入退室管理コスト削減
  - データを分散することでの災害対策

課題

情報システムの画面設計において、モックアップ (mockup) を作ることがあります。モックアップとは何でしょうか。作成する目的は何でしょうか。