usp lab.

ユニケージ　FAQ

有限会社ユニバーサル・シェル・プログラミング研究所

Universal Shell Programming Laboratory

© usp lab. 2024　Confidential Proprietary
この文書には、有限会社ユニバーサル・シェル・プログラミング研究所の機密・所有物が含まれています。
許可された者以外は使用しないでください。

FAQ作成の背景

UNIXシェルプログラミングへのイメージ

- cronやプログラム起動に使っている。あまりにもプリミティブな使い方です
- 変数、配列、関数というプログラミング言語の特性を使うと速度が上がりません
- シェルの特性を活かした使い方がされていないことが多く、そのために理解されません

We are the UNIX OS.

- UNIX哲学。時代を超えて引き継がれる技術です
- UNIXのコマンドを拡張し、UNIX OSだけで企業情報システムを作る手法です

DB業界の話題、背景

- DBMSの限界：スケールアウトやスケールアップするには高額で複雑です
- データ世界の変化：大量で、多様なデータの扱いが増えてきてます
- NoSQLの台頭：RDB（同時整合性）ではない新データ時代のDB（結果整合性）

システム開発の背景

- クラウド環境でのサービスや高速開発で自社開発が増えてます

© usp lab. 2024 Confidential Proprietary

目次

- UNIX
  - なぜUNIXなのか
  - なぜ処理スピードが速いのか
  - データ管理をどうするのか
  - バックアップはどうするのか
  - セキュリティの確保はどうするのか
  - UNIXはジョブ制御できるのか
- DB
  - DBとユニケージの違い
  - 同時整合性と結果整合性の違い
  - データのリレーションはどうするのか
  - 排他制御はどうするのか
  - ロールバック機能はどう実現するのか
  - データベースとの接続するのはどうするか
- 事例、応用、その他
  - 画像は扱えるのか
  - ウェブアプリケーションはどのように作るのか
  - IoT、SNS、ウェブサービス
  - GUI

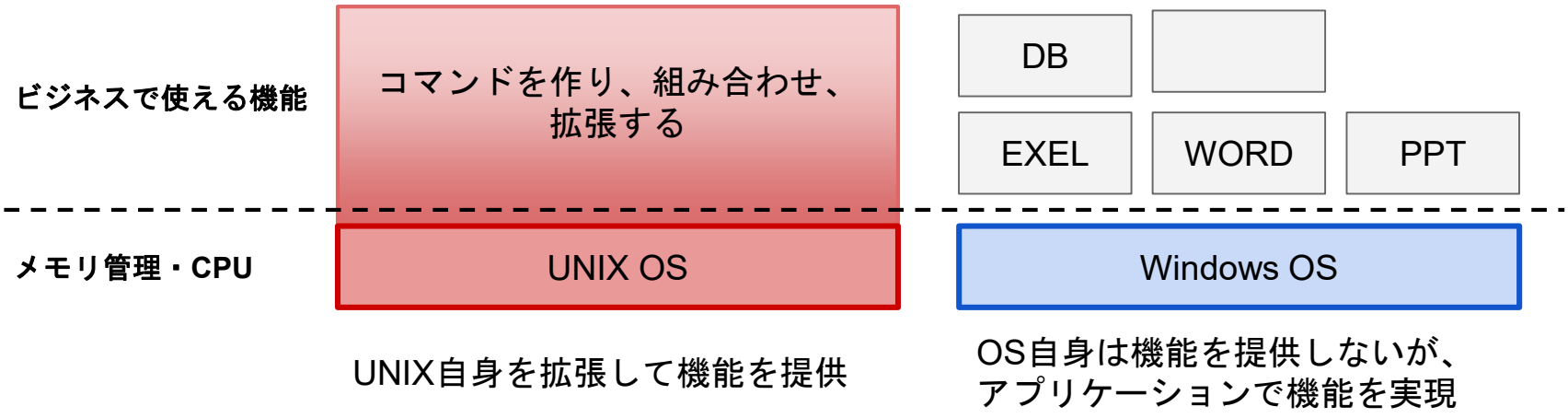
© usp lab. 2024 Confidential Proprietary

なぜUNIXなのか

usp lab.

We are the UNIX OS.

- 時代を超えて存在する技術で、その背景にはUNIXの哲学があります
- Small is beautiful. 気の利いた小さなプログラムを書き、それを組み合わせて複雑さに対処します
- UNIXは自らコマンドを作成し、機能を拡張できる自由さを備えたOSです
- UNIXは様々な種類のOSが存在し、それらを扱う企業が世界に広まり、IT業界におけるデファクトスタンダードです



Universal Shell Programming Laboratory

© usp lab. 2024 Confidential Proprietary

何故処理スピードが速いのか

usp lab.

UNIX OSのカーネル

- ダイレクトにカーネルの提供する機能を使用し、不必要な処理やミドルウェアに相当する部分の処理が無いため基本的に高速な処理系です

コマンドの工夫

- OSの拡張部品として各コマンドはプリミティブなC言語で記述され、入出力バッファ、メモリ操作、CPU操作、演算アルゴリズムが工夫されています

データ配置の工夫

- データの使用率を高める設計のノウハウがあります

正味必要な情報のデータサイズがそれを格納しているファイルサイズに占める割合

プログラミングのノウハウ

- そのコマンドの性能を発揮するための部品の組み合わせてプログラミングするノウハウと迅速な開発する手法を持っています

実例

- msort　　1,000万レコードのソートは1秒程度で実施できます
- uawk　　awkの50倍速いです

© usp lab. 2024　Confidential Proprietary

データ管理はどうするのか

- データはフラットなファイルに保存されます
- 「お作法」に基づきレベル１～レベル５に分類整理します
- データやプログラムは、ディレクトリに保存されます
- ユニケージではファイルは分類整理されているだけで、特別な管理システムやツールはありません
- データ機密性やクエリ言語による操作するミドルウェアはありませんが、OSの機能やコマンドを用いて同等の機能を提供します

データ、プログラムのディレクトリ分類例

DATA：データの分類整理
SYS：シェルスクリプト
LOG：シェルスクリプトの走行ログ
LAYOUT：レイアウトファイル
SEMAPHORE：セマフォファイル
RCV：外部システムからの受信ファイル（レベル 1)
SND：外部システムへの送信ファイル（レベル 5）
BACKUP：シェルスクリプトのバックアップ

データの分類整理

L1：原始データ
L2：確定データ ※
L3：整理データ ※
L4：アプリケーションデータ※
L5：出力データ

※空白区切のテキストファイル

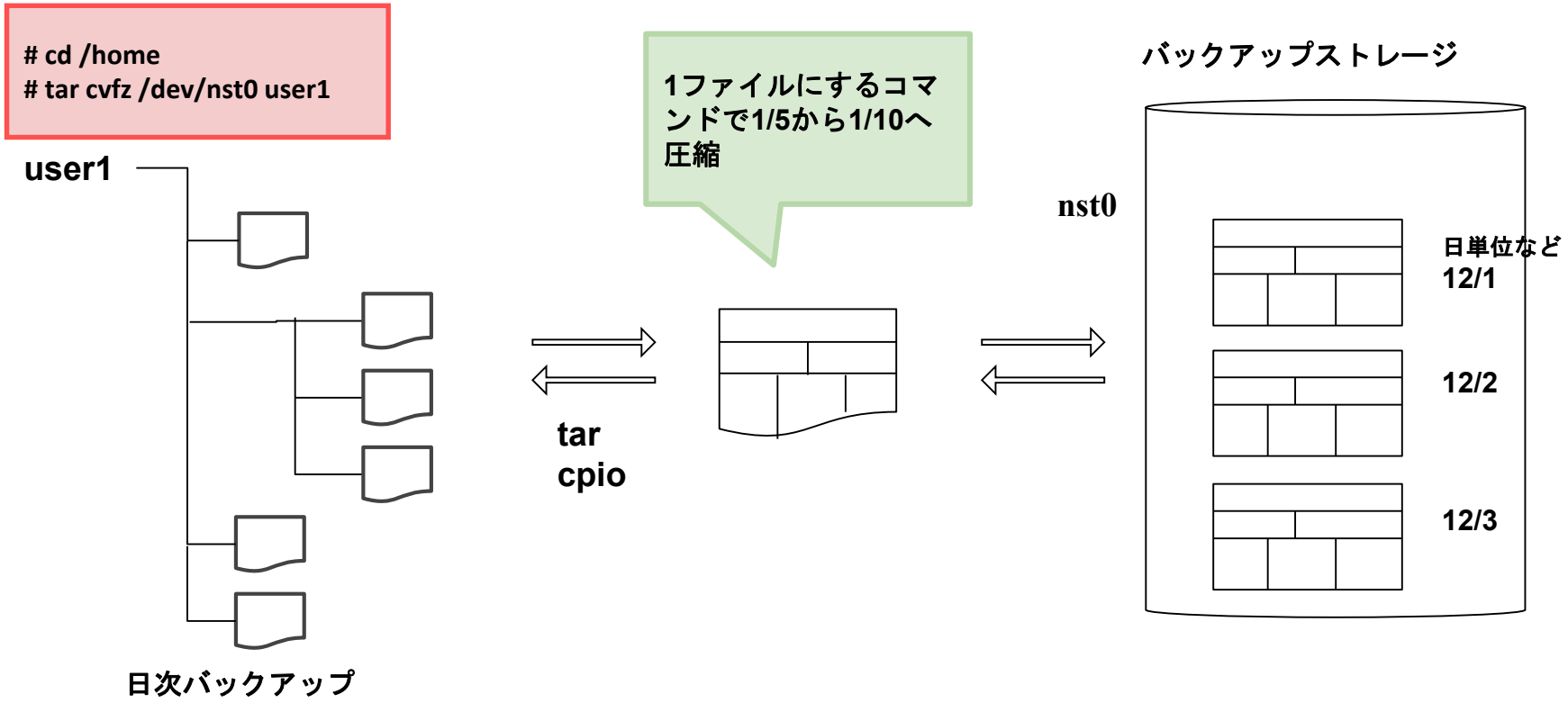
© usp lab. 2024 Confidential Proprietary

バックアップはどうするのか

- UNIXで用意されているバックアップコマンド　tar,cpio などを使います

例

/homeの下のディレクトリ user1 をディレクトリごとバックアップする場合



© usp lab. 2024　Confidential Proprietary

セキュリテイの確保はどうするのか

usp lab.

ユーザ権限

- エンドユーザごとの権限管理は権限テーブルを参照して、メニューや項目単位で読み書きの許可を与える getpermission コマンドを使います
- DBのアクセス権限のスキーマに相当します
- システム内部のアカウントは４種類あり、リソースへのアクセスを実現します
- リモートログインはアクセス経路を限定します。

システムアカウントの種類

creat	データ整理
refer	アプリケーション
user	エンドユーザ
mente	運用管理者

部長、社員、バイトなどの役職権限をgetmermission で制御

暗号化

- データは必要に応じて暗号化します

zip コマンド	パスワード付zipファイルを作成・解凍
openssl コマンド	共通鍵や公開鍵で暗号化する
gpg コマンド	GNU Privacy Guard でファイルを暗号化

高度なセキュリティの実現

- 厳しく制御できる SELinuxを使います。もしパスワードが破られても不正なソフトの混入、データ改竄、データ持ち出しを防ぐことができます
- ID、パスワード、ワンタイムパスワード、２アクセス認証、生体認証、外部機器の接続の制限でセキュリティを確保します

© usp lab. 2024 Confidential Proprietary

UNIXはジョブ制御できるのか

UNIXはマルチユーザ、マルチタスク OSであり、複数のユーザの複数のジョブを同時に管理できます

ジョブ起動時に、&(バックグランド)を指定して並列処理	**&(バックグランド)**
処理の途中でジョブを並列化したり、順次化の処理	**bg, fg コマンド**
並列処理の優先順位の変更	**nice コマンド**
現行のジョブの停止や中断	**stop, kill コマンド**
コマンド　並列で動作している状況の確認	**jobs, ps, tree コマンド**
CPU, メモリ, ディスクの使用制限	**cgroupで設定**

© usp lab. 2024　Confidential Proprietary

RDBとユニケージの違いとは

usp lab.

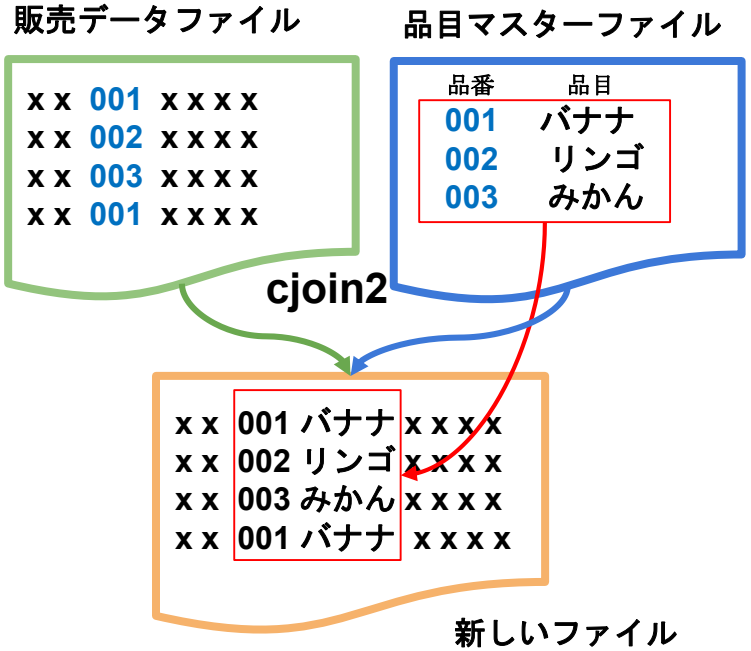
	RDB	ユニケージ
整合性方針	同時整合性	結果整合性
更新ロック	関係するすべてのプロセスとデータ(レコード)をロックする（高負荷＋負荷はデータ量の累乗に比例する）	更新対象のファイルのみロックする（低負荷＋負荷はデータ量に線形比例する）
情報鮮度管理	不要（同時性が担保されているため）	必要（ファイル更新タイミングを意識した設計開発）
データ履歴	データは上書きし、トランザクションは非保持	処理結果は別ファイルを出力。トランザクションは保持される
ロールバック	メカニズムが存在	データ・トランザクションとも履歴があるので、特別なメカニズムは不要（アプリケーションで実装）
スケーラビリティ	スケールアップ型	スケールアップ型＋スケールアウト型

© usp lab. 2024 Confidential Proprietary

データのリレーションはどうするのか

- join1（インナージョイン）、join2（アウタージョイン）などのコマンドを組み合わせます
- 必要な時にその場でファイル同士を結合します
- データのモデリングが柔軟に表現できます
- データの結合が簡単にでき、キーバリュー形式で保存できます

例



ユニケージコマンド
結合の種類別にコマンドが用意されている

結合キー	インナージョイン	アウタージョイン	マルチジョイン	フルジョイン
ソート済	join1	join2	join1x join2x	loopx
ソート未済	cjoin1	cjoin2	cjoin1x cjoin2x	

© usp lab. 2024 Confidential Proprietary

排他制御はどうするのか

usp lab.

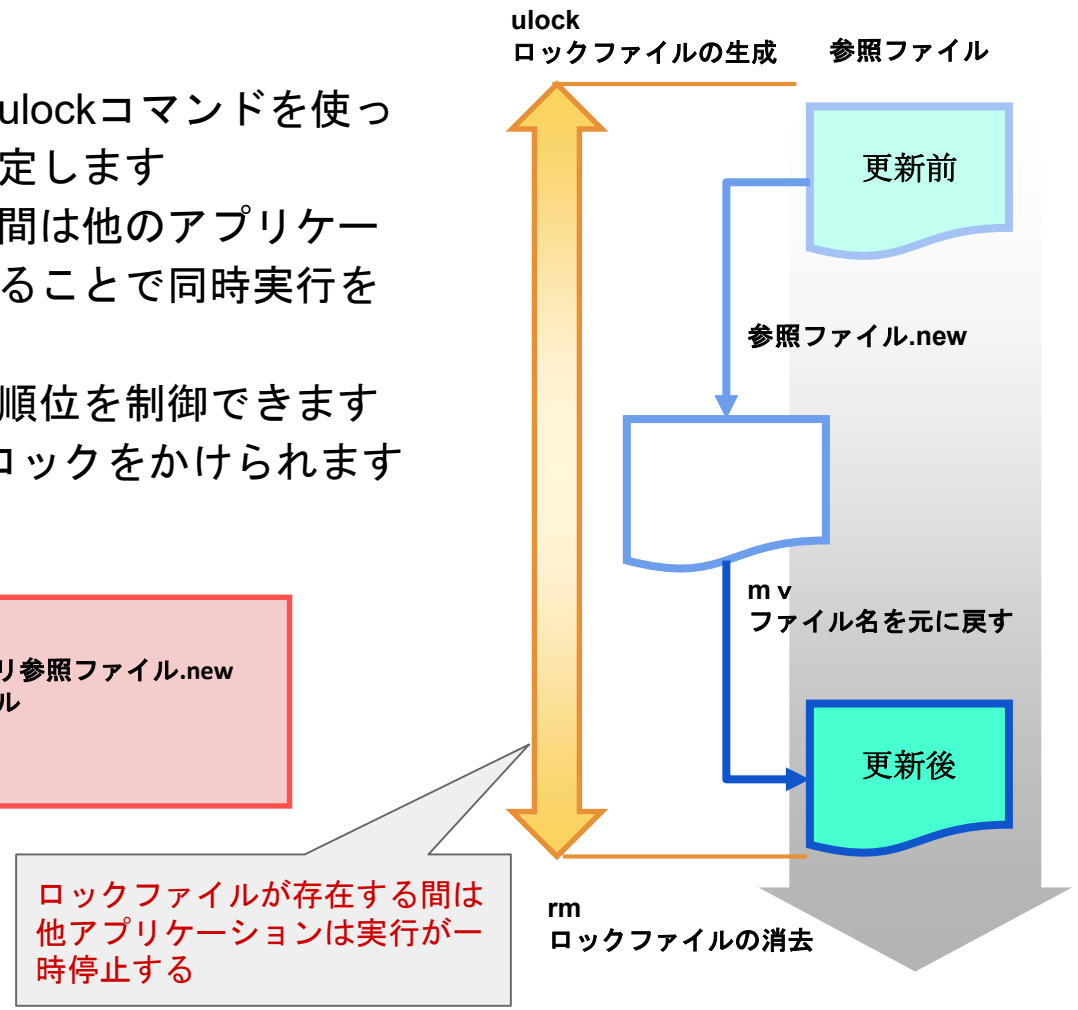
ユニケージの実装

- 排他的に処理したい範囲をulockコマンドを使って明示的にプログラムで指定します
- ロックファイルが存在する間は他のアプリケーションは実行が一時停止することで同時実行を制御します
- アプリケーション側で優先順位を制御できます
- 1ファイル＝1レコード でロックをかけられます

シェルスクリプトのイメージ

```
if ulock ロックファイル then
   upl 入力トラン　アプリ参照ファイル　>　アプリ参照ファイル.new
   mv アプリ参照ファイル.new　アプリ参照ファイル
   rm ロックファイル
fi
```

※ulockコマンドには
タイムアウト、強制ロックファイル消去
機能があります



Universal Shell Programming Laboratory

© usp lab. 2024　Confidential Proprietary

データベースと接続するには

usp lab.

DB接続用コマンド

- 引数や設定ファイルにサーバー名、データベース名、ユーザー名、パスワード、SQL を指定して起動します
  - sqlplus　　ORACLE
  - tsql/sqlcmd　　SQLServer
  - mysql　　MySQL
  - psql　　Postgres

ファイル入出力

- CSV形式などのファイルで出力/入力する
  - SQLLDR　　ORACLEが提供しているデータアップロード製品
  - bcp(bulk insert)　　MSSQL を実行できるコマンドを使用する

```
$ cat test.sql
  select * from table;
  exit 0;
$ sqlplus -S scott/tiger@myhost:1521 @test.sql
```


© usp lab. 2024　Confidential Proprietary

画像は扱えるのか

- 画像のバイナリデータをファイル単位で管理します
- UNIXで公開されている画像処理コマンドを利用します
- **ImageMagickコマンド**　画像を拡大縮小、切出、結合や形式変換などを行えます

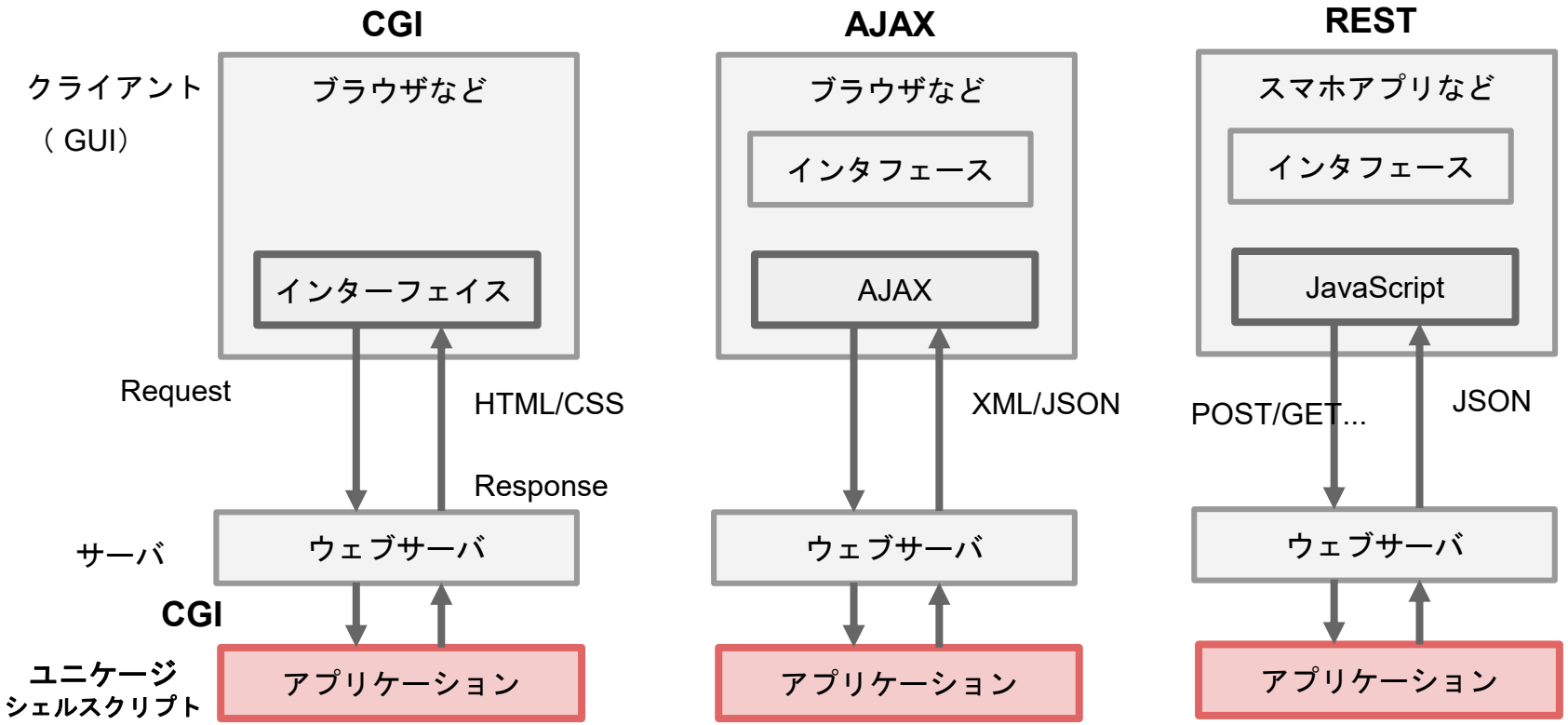
画像形式の変換	convert ファイル 変換後ファイル.拡張子
画像の拡大縮小	convert -scale 比率% ファイル 変換後ファイル convert -scale 幅x高さ ファイル　変換後ファイル
画像のモノクロ化	convert ファイル名 変換後ファイル.pgm
部分画像の切り出し	convert -crop 左上x座標,左上y座標+幅+高さ ファイル 変換後ファイル
複数画像の結合	montage 入力画像1 入力画像2 入力画像3 以下入力画像… 出力画像名
アニメーションGIF	convert 入力画像1 入力画像2 以下入力画像… 出力画像名.gif

© usp lab. 2024　Confidential Proprietary

WEBアプリケーション（GUI）はどのように作るのか

usp lab.

- ユニケージはサーバ側のアプリケーションを担います
- 各種 UIクライアントとウェブサーバを通じて接続します
- REST などのWEB-API を提供することもできます



© usp lab. 2024 Confidential Proprietary

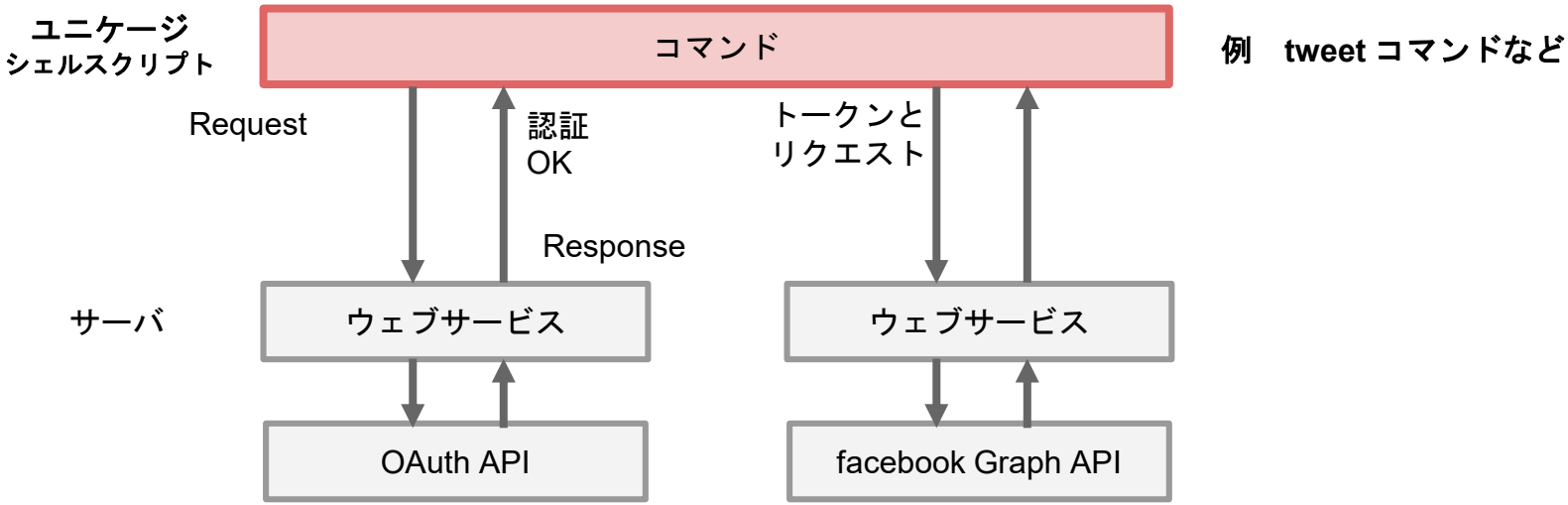
SNS,AI,Webサービスはどう利用するのか

usp lab.

- twitterやfacebookのSNSやgoogleなどのWebサービスと連結できます
- ユニケージはクライアント側のアプリケーションを担います
- 公開している認証の仕組みや提供する APIを利用するコマンドを提供します

例

- OAuthやOpenIDでサーバにアクセスする権限を認証します
- 認証後、アクセストークンを使って各サーバのAPIを使い、リクエストします
- データを取得し、処理をして結果を返します



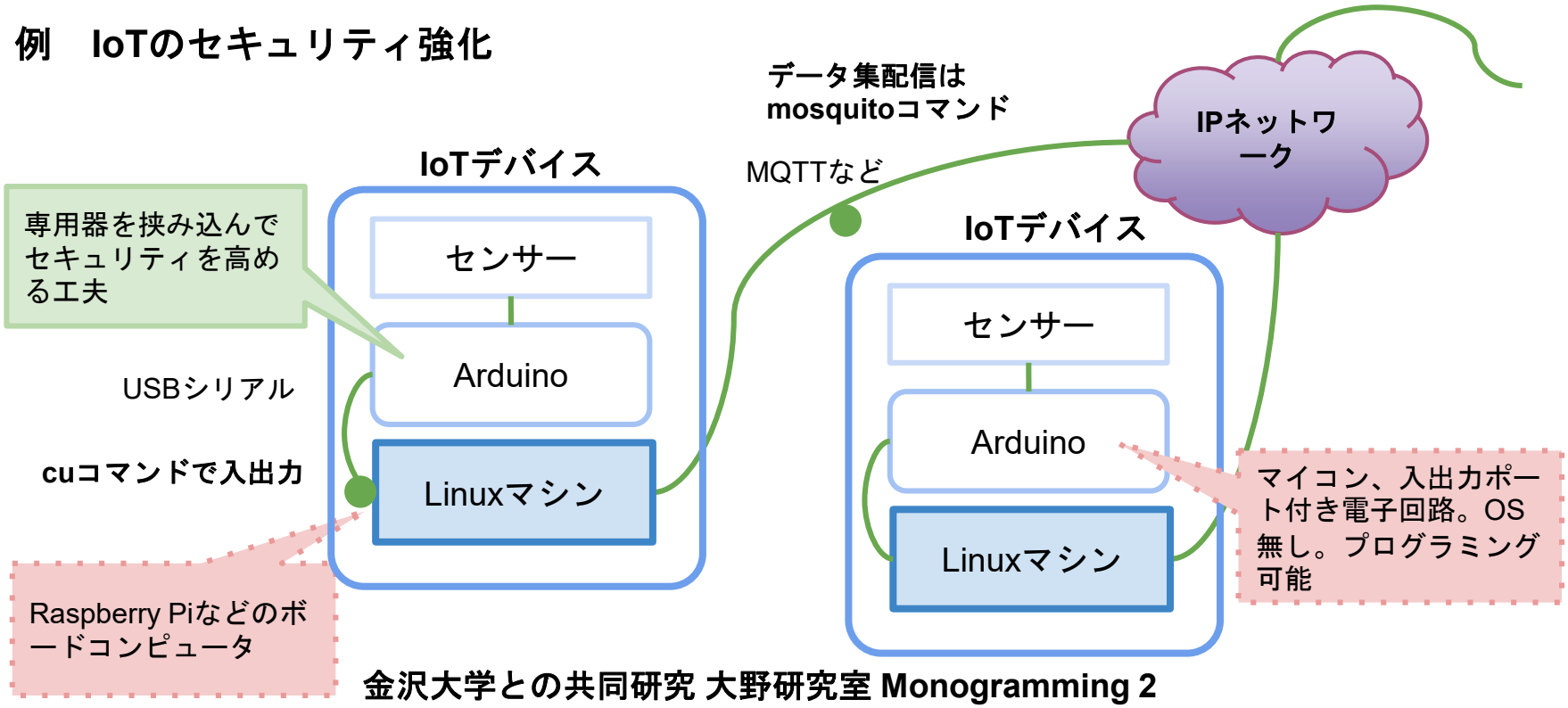
Universal Shell Programming Laboratory

© usp lab. 2024　Confidential Proprietary

IoTの実現はどうするのか

- IoTデバイス(RasberryPi)はLinux OSを搭載しユニケージコマンドが使えます
- Arduino などのNon-Intelligenct 回路を挟み、デバイス操作を単純化します
- SELunixを適用し、不正操作からIoTデバイスを守ります

例　IoTのセキュリティ強化



金沢大学との共同研究 大野研究室 Monogramming 2

© usp lab. 2024　Confidential Proprietary