

2005年度下期未踏ソフトウェア創造事業

# ネット公開を目的とした マルチウィンドウアプリ用フレームワーク

## 成果報告

Ruby/TkORCA

九州工業大学情報工学部知能情報工学科 永井 秀利

# 開発物の名称と目的

- 正式名称 Ruby/Tk On Remote-Frame-Buffered Canvas  
( Ruby/Tk On RFB Canvas )
- 通称(略称) **Ruby/TkORCA** [ ルビー・ティーケー・オルカ ]
- 開発目的  
ローカルのウィンドウシステム上に限られていた GUI プログラミング技術の対象領域を, 広くネットワークアプリケーションにまで拡張する
- コンセプト  
**「 どこでもGUI ! 」 ( GUI, Anywhere ! )**

Ruby/TkORCA

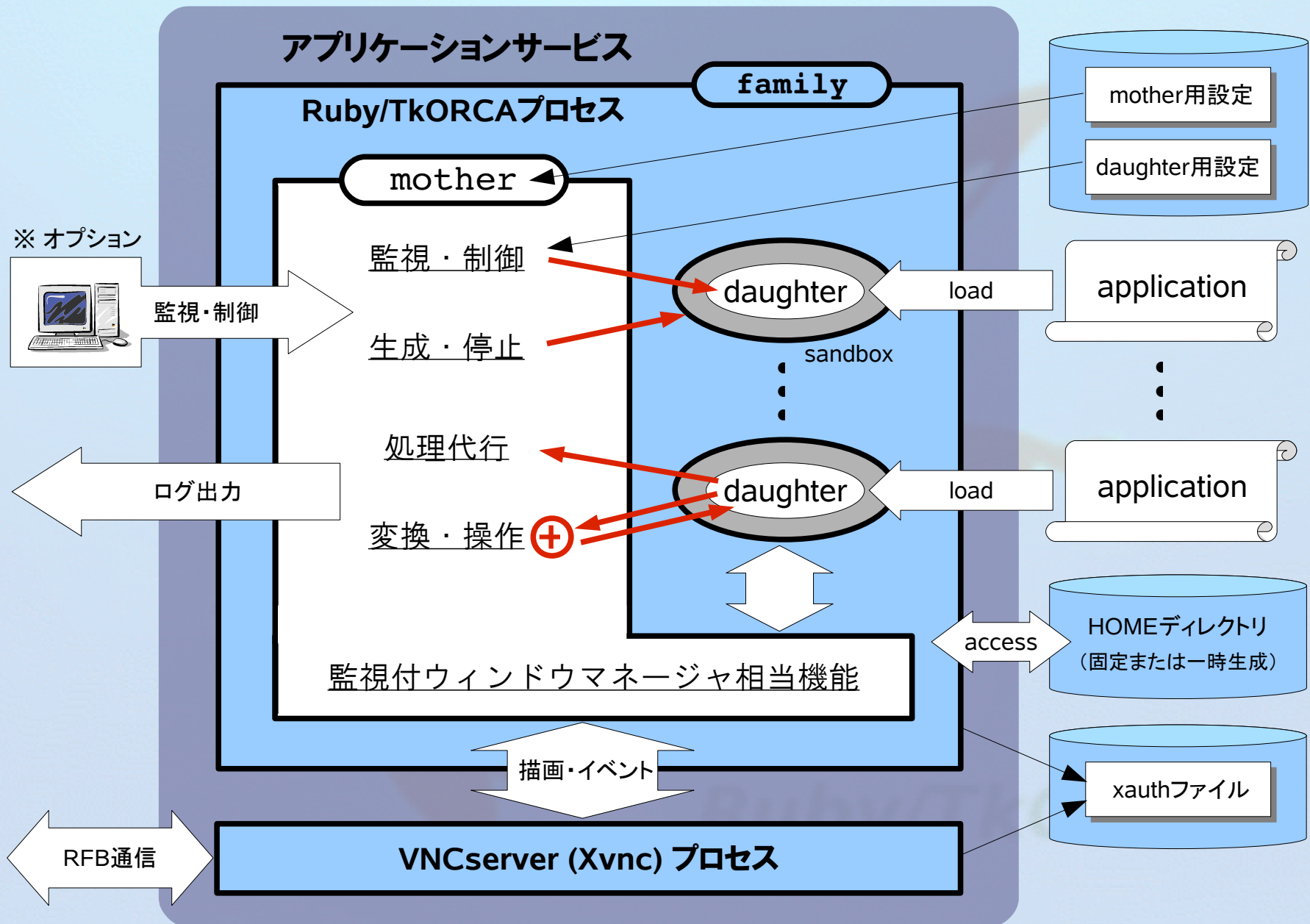
# 現状でのネットGUIアプリ作成時の問題点

- 基本傾向として、能力が高い枠組ほど制約も厳しく、想定範囲を越えようとするとも実現が非常に困難だったり不可能だったりが多い
- Flash等の独自フレームワーク
  - 開発者に**特有の技量を要求**
  - インストールされたクライアント用プログラムのバージョンに依存するため、**適用可能範囲が狭く、発展や機能改良も鈍重**
- Ajax ( Asynchronous JavaScript + XML )
  - サーバ/クライアントの連携やクライアントの差異への**配慮が面倒**
  - 機能向上に対し、**開発コストや情報漏洩リスク等が急激に増大**
- VNC ( Virtual Network Computing )
  - 接続を許可した**相手を信用し過ぎ**ており、監視や制御ができない
  - 不特定多数をサービス対象とする**仕組みが欠落**している

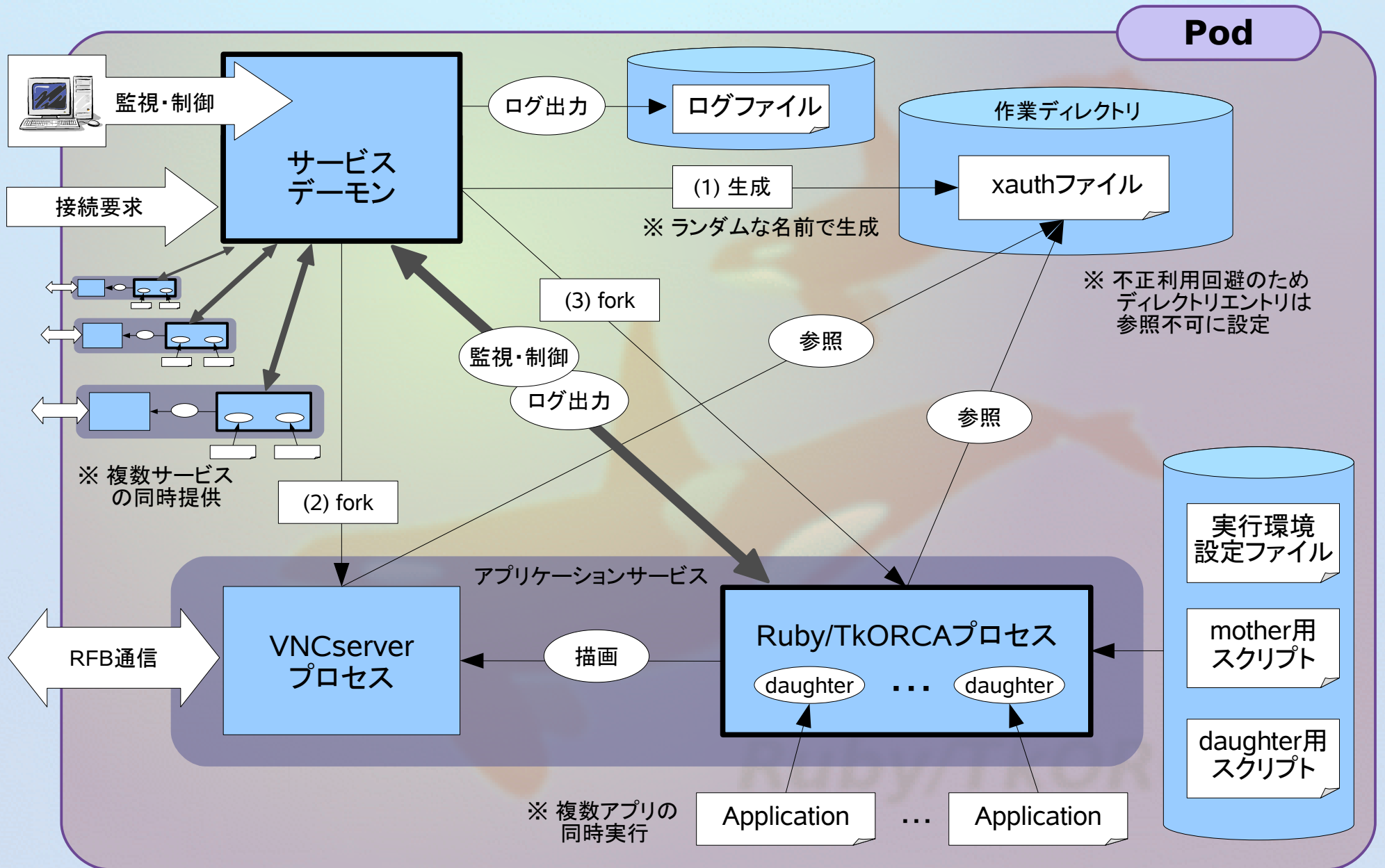
# Ruby/TkORCAの思想

- VNCの利点を活かしつつ、その問題点を回避した枠組とする
  - 高度なインタラクティブ性や、プログラムやデータをクライアントに送らないことによる低い情報漏洩リスクなどの利点はそのままに享受
  - 問題の源になる一般的ウィンドウマネージャは動かさず、相当機能の実装によりウィンドウ操作も監視や制御の対象とする
- 不特定多数の他者へのサービス提供に利用できるものとする
- 開発者に対して、特有で特殊な技量を要求しない
- ローカル用とネット用とでソース変更をほとんど必要としない
- 監視や制御の機構をあらかじめ組み込んでいなくても、アプリケーションの監視や制御を可能に(改良・強化も容易に)する
- 援用する他のソフトには一切手を加えない
  - 導入コスト低減や、他ソフトの強化・改良からの最大限の受益のため
  - 広く使われているものに「似て異なるもの」を開発する愚は避ける

# Ruby/TkORCAの構造



# Ruby/TkORCAサーバの構成概略図

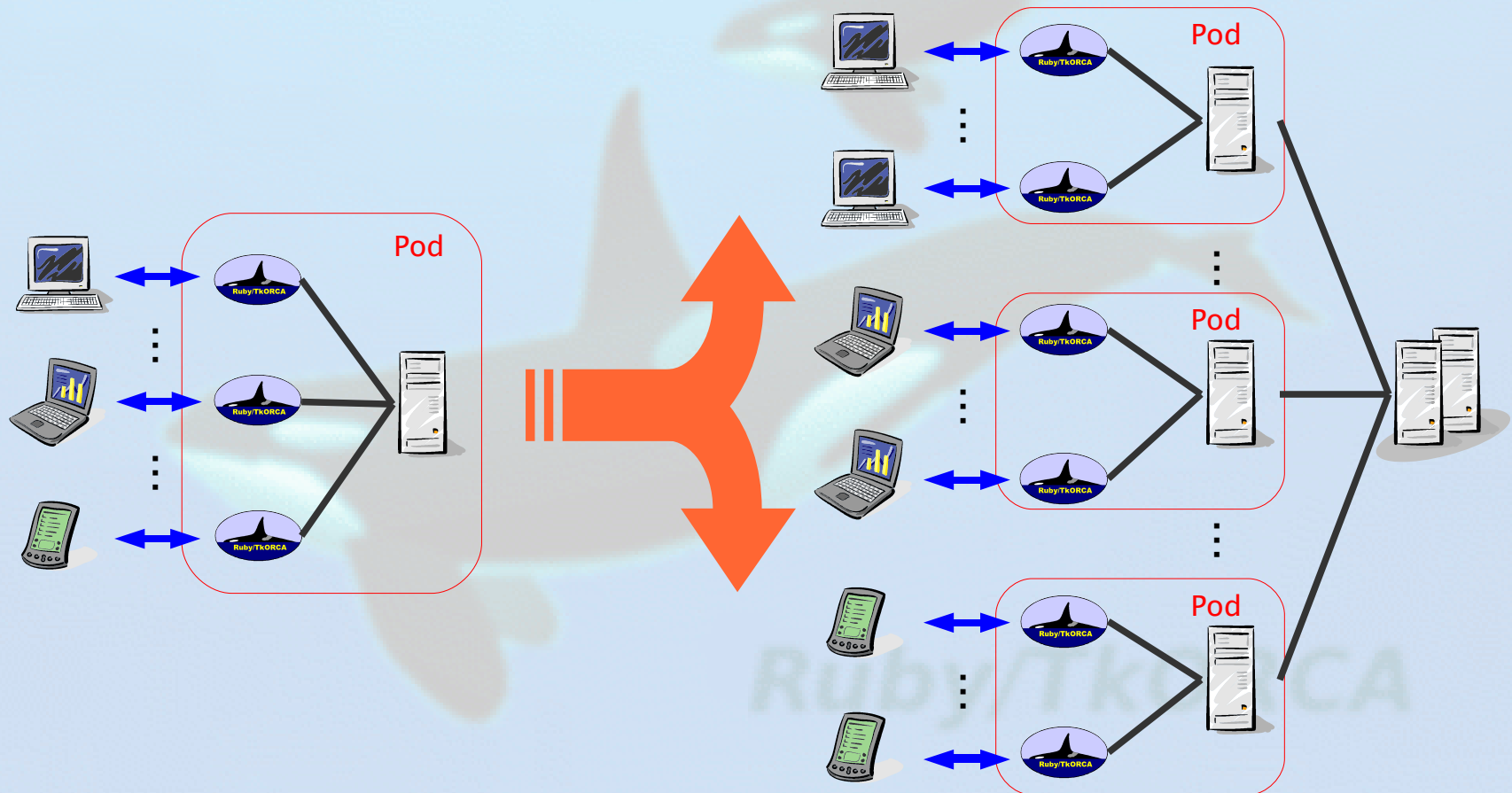


# Ruby/TkORCAの特長

- **インタラクティブィ**
  - ローカルGUIと同等のものを獲得可能
- **ポータビリティ**
  - Ruby/Tk (Tcl/Tk) ベースであり, 多くのOSでソース互換
  - アプリケーションは同じソースでローカルからネットワークまで
  - クライアントはPCからPDAや携帯クラスまで  
(何らかのVNCviewer相当品が動作すればサービス利用可能)
- **スピーディ**
  - 開発や公開における低い作業コスト
  - 特殊なクライアントを必要としないことなどによる即応性の高さ
- **スケーラビリティ**
  - トイプログラムから大規模アプリケーションまで
  - 1台への提供から, 巨大バックボーンを持つ大規模サービスまで

# サービスのスケールアップ

- Ruby/TkORCAが規定するのはアプリケーションの表層のみ
- バックエンドを強化することでサービス規模の拡大が可能





# Ruby/TkORCAにおける監視制御機構

- **マスター・スレーブコントロール**
  - 各アプリケーションはそれぞれ個別のdaughter上で動作し, motherによって監視や制御がなされる
- **パッシブコントロール**
  - ログ出力の内容などに基づき, 受動的に制御機構を起動できる
- **アクティブコントロール**
  - アプリケーションの動作状態とは無関係に, サーバ側からサービスプロセス, さらにはアプリケーション内部に直接干渉できる
- **イミディエイトコントロール**
  - クライアント側のクリックなどの操作を待つ必要なく, 制御内容の反映やメッセージ表示を即時に適用できる

# Ruby/TkORCAにおけるセキュリティ確保

- Ruby/TkORCAの**サービス利用許可**
  - デフォルトでは無条件許可（不特定多数へのサービスのため）
  - 必要ならVNCserver (Xvnc) の認証機構を利用
- 同一サーバ上の**サービス間干渉の回避**
  - 読出し不可ディレクトリ上のランダム名称のHOMEディレクトリやxauthファイルを利用
- 同一サービス上の**アプリケーション間干渉の回避**
  - RubyおよびRuby/Tkの機能を活用したsandboxによる分離
- 監視・制御のための**サーバログイン許可**
  - ローカルホストからの接続のみを許可するパスワード認証
- **通信経路上のセキュリティ確保は対象外**
  - 「VNC over SSH」のような一般的手法の利用を想定
  - 必要ならRuby/TkORCAで接続元アドレスでの許可/拒否設定は可能

# 公開用アプリケーションの必須要件

- GUIの最表層がRuby/Tk または Tcl/Tk であること
- 画面表示しないものであれば, どのようなライブラリやプログラムを使っている構わない
- 画面表示をするものでも, 何らかの手段でTkのコンテナウィジェットに埋め込めれば利用可能  
(Tkの歴史の長さにより, Tkへの埋め込み機能を持つライブラリも多い)

## 必要とされる技量

- Ruby/Tk または Tcl/Tk でスクリプトが書けること
- 個人利用に限定しないのなら,  
「受け取った文字列を無条件にevalしない」程度の分別はあること

# Ruby/TkORCAでの公開アプリ開発

## 1.5+ $\alpha$ ステップの低コストな流れ

- S=1.0:** Ruby/Tkを使ってローカルのウィンドウシステムで動くようにアプリケーションを作る(または動くものを持ってくる)
- S+0.5:** アプリケーションをRuby/TkORCAに読み込んで, ローカルウィンドウの上での実行テスト  
もし動かなければ, 以下を試みる
- a) daughterのセキュリティ制約を緩める  
(デフォルトは厳しすぎるくらいの設定なので, 適切な程度に緩める)
  - b) motherでチェックするように, 動かない部分のwrapperを書く
  - c) 動かない部分のソースに手を入れる
- S+ $\alpha$ :** ローカルテストに合格したら, サーバに公開登録して完了

Ruby/TkORCA

# Ruby/TkORCAシステムの必要物

- Ruby/TkORCA自体はpure Ruby/Tkで構築
  - コンパイル等の必要なく, 導入可能
- 最低限必要なもの
  - a) inetd対応のXvnc (標準入出力を通信に使えるもの)
  - b) (a)のXvncに出力可能なTcl/Tk
  - c) (b)のTcl/Tkのライブラリを使うRuby/Tk  
(ただし, Ruby 1.8.5以降 or それに含まれるRuby/Tkをバックポートしたもの)
- その他, アプリケーションの必要に応じて
  - xfs, kinput2, Mesa, Tk拡張ライブラリ, ...

**重要点: 「特殊なバージョン(パッチ)は不要」**

共存問題なし, 利用可能範囲が広い, 必要物の改良を即座に受益, など利点大

# サポート範囲

- Unix系OS
  - 現状での公式サポート対象
- Windows
  - Windows nativeのVNCserverは、OS稼働中の画面をそのまま送り出してしまうため、サポート対象外
  - 「Cygwin版Xvnc」 + 「Cygwin X用Tcl/Tk」 + 「同Tcl/Tkを利用できるRuby/Tk」 + 「Ruby用のfork等互換ライブラリ」の組合せなら多分利用可能だが、Cygwin特有の問題の検証(と必要なら修正)が未完のため、非公式サポート(将来は公式化の予定)
- MacOS X (MacOS 9以前はWindows nativeと同様の理由で対象外)
  - X window system用に構築したXvncやTcl/Tkを使えば動くはずだが、検証未完のため非公式サポート(将来は公式化の予定)
- その他, Xvnc相当とRuby/Tkが動く環境
  - 無保証だが、十分に稼働可能と推定

# デモンストレーション

- C言語で作られた外部ライブラリを使用し, コンソール出力もするようなアプリケーション
- OpenGL (Mesa) を用いているアプリケーション
- 複数アプリケーションの同時サービス
- 5分で作るネットワークGUIアプリ

・・・など, 時間が許す限り

( 間に合わない分は後でデモスペースで実演できれば・・・ )

Ruby/TkORCA

## まとめ

- 「ネットで使えるGUIアプリが簡単に作れたらなあ」と少しでも考えたことがあれば、ぜひRuby/TkORCAをお試してください
- 例えば...
  - ちょいちょいと作ったツールを手間をかけずにリモートアプリに
  - 研究室などのGUIを使ったデモアプリを、特殊なライブラリや機密情報を一切見せることなくそのまま外部公開に
  - マルチウィンドウ上のドラッグ&ドロップでお買物. 臨時ニュースや変動もサーバ側から即座に反映できるショッピングサイトに
  - ゴミと化していた旧式機を復活させるthin client環境構築に
  - 軽量長時間駆動の店内端末に. お客様を移動させない, 待たせない. もし端末を盗まれても, 情報までは盗まらない

... などなど
- 「ネットワーク」+「GUI」というキーワードが当てはまる様々な分野にご活用いただけますと幸いです