

「ファンタシースターユニバース」の開発現場より  
～5年間の歩み～



株式会社セガ 第3GE研究開発部  
デザインセクションマネージャー 酒井 智史  
プログラムセクションマネージャー 節政 暁生

# 「PHANTASY STAR UNIVERSE」とは？



『PHANTASY STAR UNIVERSE』

**8月31日発売**

PS2 / Windows

¥6,800(税込¥7,140)

※Xbox360版も発売予定

# 「PHANTASY STAR UNIVERSE」とは？

「ストーリーモード」と「ネットワークモード」の2つの柱を持つRPG

## ◎ストーリーモード

- ・主人公「イーサン・ウェーバー」の成長物語(王道SFファンタジー)
- ・30時間以上のプレイ時間
- ・アクションバトルはネットワークモードと同じシステム
- ・エクストラモードでネットワークモードの疑似体験

## ◎ネットワークモード

- ・最強のキャラクタークリエイト
- ・ロビー(MMO)とフィールド(MO)の融合
- ・マイルームでの合成、商売の機能で「生活」ができるゲームに。
- ・進化したコミュニケーションシステム

二つのモードは同じ世界でストーリーは密接に繋がっている！  
ゲームシステムが同じなので、違和感なくネットワークに行ける！

# 「PSO」から「PSU」への歴史

2000年12月21日	「PHANTASY STAR ONLINE」	DC	
2001年 6月17日	「PHANTASY STAR ONLINE Ver.2」	DC	
2001年12月20日	「PHANTASY STAR ONLINE」	PC	韓国、台湾進出
2002年 9月12日	「PHANTASY STAR ONLINE EPISODE I&II」	GC	エピソード追加 マルチモードの搭載 ユーザーの低年齢化
2003年 1月16日	「PHANTASY STAR ONLINE EPISODE I&II」	Xbox	ボイスチャット
2003年 3月頃	<b>「PHANTASY STAR UNIVERSE」開発開始</b>		
2003年11月27日	「PHANTASY STAR ONLINE EPISODE III C.A.R.D Revolution」 「PHANTASY STAR ONLINE EPISODE I&II Plus」	GC	EP3は一部 サーバークライアントに
2004年 7月 8日	「PHANTASY STAR ONLINE Blue Burst」	PC	サーバークライアント型へ キャラクターのサーバー保存
2005年 2月 3日	「PHANTASY STAR ONLINE Blue Burst EPISODE IV」	PC	エピソード追加
2006年 4月 20日	<b>「PHANTASY STAR UNIVERSE Closed β trial」運営開始</b>	PC	ネットワークトライアルテスト
2006年 8月 31日	<b>「PHANTASY STAR UNIVERSE」発売！</b>	PC、 PS2	完全新作

# 「PSU」の開発について・1

## ◎3年半の開発期間、最大約80名の内部開発人員

- ・最初の一年は、企画とデザインイメージだし、ライブラリ選定の期間。
  - ＞イメージ統一のため、イメージムービーを作成
- ・ストーリーモードが先行して開発、ネットワーク、サーバー部分を後から作成

## ◎2つのゲームを同時に作り上げると言う問題

- ・初期には、2つのモードで共有できる部分が多いので、1.5倍程度の手間で出来るのではないかと思っていたが、実際には2倍以上の手間がかかってしまった。
  - ＞ストーリーモードのボリュームに苦しめられた。(イベントムービー3時間半)
  - ＞ネットワークモードには、ゲームシステムの複雑化と、多人数化によるバグに苦しめられた。(ダミークライアントによるテストは効果的だった。)

## ◎3つのプラットフォームへの対応

- ・PC版での開発を優先して行ったため、PS2のメモリの制約に苦しめられた。
- ・そのためPC版のクオリティアップも困難であった。
- ・Xbox360への移植は比較的容易に行うことができた。

# 「PSU」の開発について・2

## ◎開発の効率をあげるための色々な試み

- ・プロジェクト全体……alien brain  
企画、デザイナー、プログラマー全員がデータを同じ場所で共有化できた。  
バージョン管理ができるため、複数人でのファイル共有には非常に有効。  
画像掲示板的使い方をして、デザインの意見交換の場としても使用。
- ・デザイナー……3D STUDIO MAX  
膨大な物量进行处理するためにSIから移行。  
豊富なフリープラグインなどにより、モデリング効率は格段にアップ  
モーションもすべてMAXで行った……CharacterStudio の活用
- ・プログラマー……IncrediBuild  
ソースの巨大化による、プログラムのコンパイル時間の長さが問題になったために導入。  
10倍から20倍の短縮ができた。
- ・バグチェック……B-TRAQ  
担当者に直接メールが行く事、全員でリアルタイムに情報共有が出来たことで、  
バグチェックのレスポンスが大幅に向上した。
- ・その他内製ツール……3機種のエクスポーター  
イベントエディター、エフェクトエディター  
マップエディター

# 「PSU」のゲームデザイン・1

## ◎オフラインRPGとして……

- ・ストーリーモードとネットワークモードのシステムの共有化が大前提。
  - ＞世界観とシステムとしてのチュートリアルになるように。
- ・章立てのストーリーモード
  - ＞区切りのつきにくいRPGへのアンチテーゼ
  - ＞30時間程度のプレイ時間で「長すぎない」ものを。
  - ＞あちこちの星に舞台が飛ぶガーディアンズと言う制約を解消。
- ・派手さと爽快感のある戦闘システム
  - ＞右手と左手の武器を持ちかえるように。
  - ＞MPの概念をなくし、武器に持たせるPPのシステム。
  - ＞戦闘タイプ概念によって、職業固定の制約を排除。
  - ＞「フォトンアーツ」のシステムにより、個性と爽快感、レベルアップの楽しさを付与。

→ネットワークモードのシステムとも違和感なくつながるものに。

# 「PSU」のゲームデザイン・2

## ◎ネットワークRPGとして……

### ・MO部分(フィールド)とMMO部分(ロビー)の融合

- ＞手軽なネットワークRPGの部分と、たくさんの人とのコミュニケーションの両立
- ＞ロビー＞フィールド＞ロビー＞フィールドのシステムで独自の「目的地を目指す」形に。

### ・チャットのライブ感とキャラクターへの愛情の増強

- ＞吹き出しチャットの改良……吹き出しの種類、吹き出し合体などの新システム。
- ＞キャラクターの顔を出す「カットインチャット」
- ＞キャラクタークリエイトをさらに細かいものに改良。(顔モーフィング、着替え)

### ・マイルームとパートナーマシナリーのシステム

- ＞プレイヤーがログインしていなくても残る「マイルーム」
- ＞生産、店番などをパートナーマシナリーが行う。

→ストーリーモードと世界観、キャラクターを共有する事による世界への没入感

- ＞「なんでもアリ」だからSFファンタジー。



# PSO > PSOBB > PSUの違い

	PSO、PSOEP2	PSOBB	PSU
通信方式	サーバー & P2Pハイブリッド型 ※クライアント同士はUDPで通信	完全サーバークライアント型	完全サーバークライアント型
セーブデータ	クライアント	サーバー オンライン専用	・オフ=クライアント (PC版サーバーセーブ対応) ・オン=サーバー
ゲーム内の各種処理の対応方法	各種処理はクライアント側で対応	ゲームの主要処理(ショップ、アイテム、エネミー等)をサーバー側で判定	ゲームの処理の殆どをサーバー側で判定 ある程度の同期処理
帯域	ナローバンドが主体	1Kbps程度 ブロードバンド推奨	10~15Kbps程度 ブロードバンド専用
パッチ	なし(Ver.2ディスクにてバグ修正)	あり	あり PS2はメモリーカードに PCはEXEを更新
備考	運営費を安く上げる事を意識	チート対策に重点	MMOシステムとしての安定性を重視

# PSOとPSUのゲームデザイン上の工夫

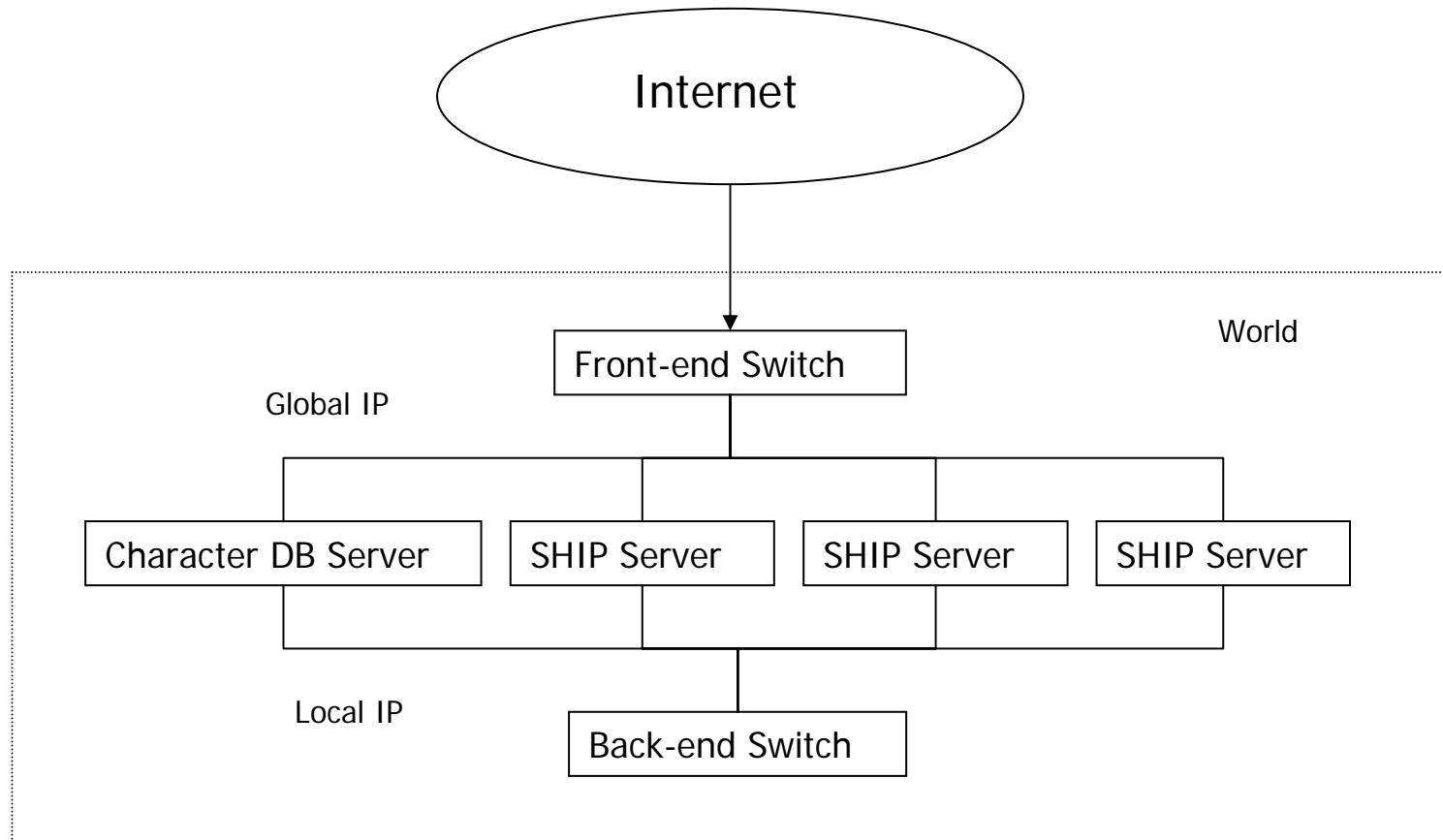
## PSOの場合

- ・歩きパケットの節約
- ・非同期ゲームでレスポンスの向上
- ・必ずロックオンカーソルで攻撃対象を決定
- ・エネミー座標の同期を取らない
- ・小さい部屋に区切って一つずつクリア
- ・ボスだけ行動の同期を取る

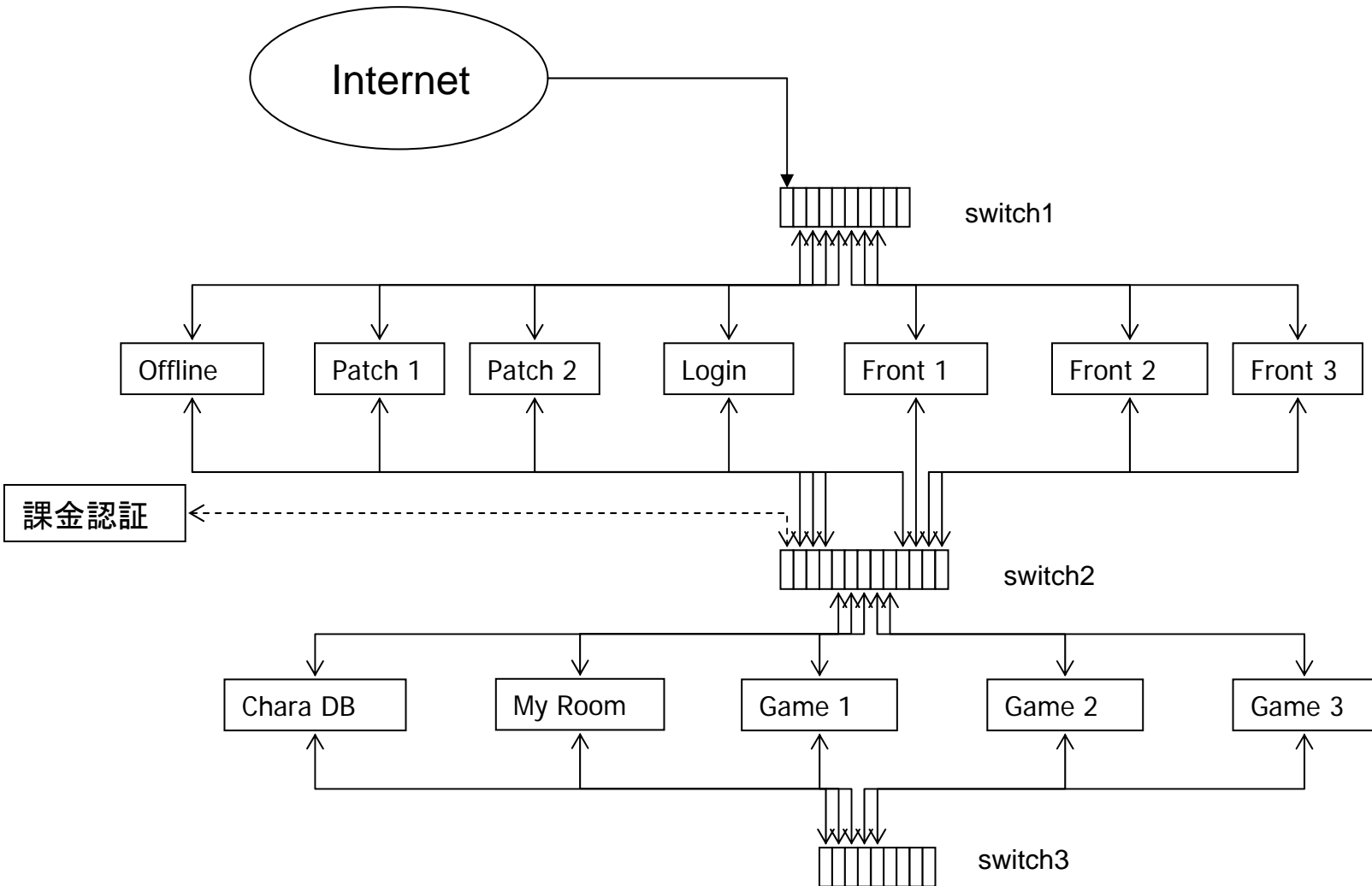
## PSUの場合

- ・更なるアクション性の向上
- ・サーバーサイドで戦闘計算等処理
- ・レスポンスの良さと堅牢性はトレードオフが必要
- ・エネミー行動や座標の同期処理
- ・エネミーがプレイヤーの画面外に出た時につじつまを合わせる

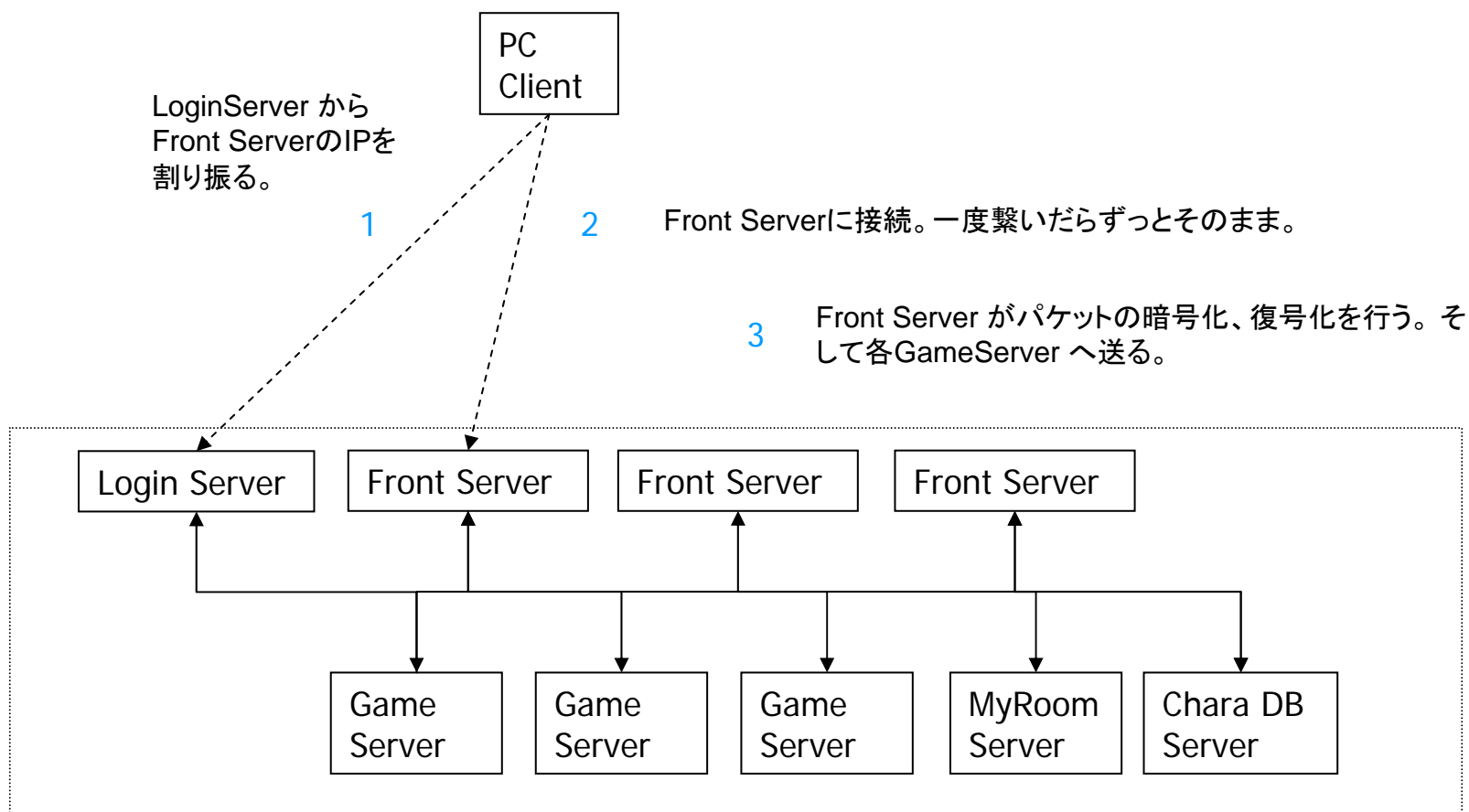
# PSO BBサーバー物理構成



# PSUサーバー物理構成



# サーバーへのアクセス



# サーバーを分ける意義

## メリット

- ・SSL暗号を使う場合、セッションを張る時の重い処理を分散出来る。
- ・ゲームサーバーの処理が軽くなる。
- ・マップ移動等をシームレスに行わせる。
- ・インターネットから直接ゲームサーバーが見えなくなり、ハッキングされにくくなる。

## デメリット

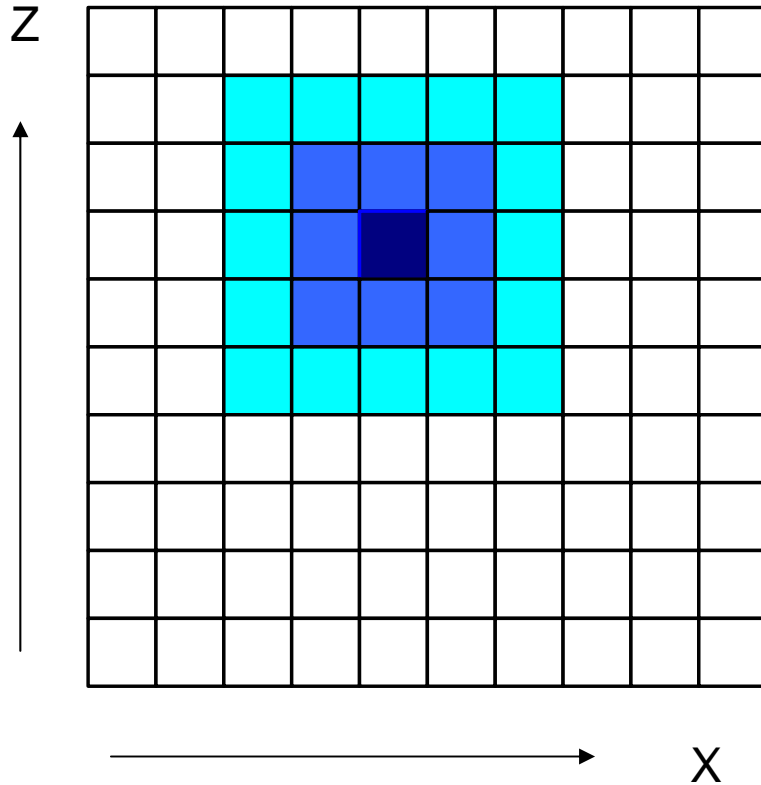
- ・必要なサーバー機器が増える
- ・レイテンシーが若干悪くなる
- ・必要なプログラム量が増える。
- ・サーバー間の同期処理が面倒。

# PSUのパブリックエリアとプライベートエリア

- PSUではMMOであるパブリックエリアとMOであるプライベートエリアでシステムが分かれている。
- MMOとMOのいいとこ取りのようなスタイル

パブリックエリア	プライベートエリア
シティ 中間ロビー	フィールド マイルーム
数百人(MMO的)	6人(MO的)
リアルタイム性の少ない処理に向くよう設計	リアルタイム性の強い処理に向くよう設計
負荷分散のため、パケットを制限する仕組みが入っている。	戦闘計算や攻撃判定など主要な処理がサーバ側で行われる。

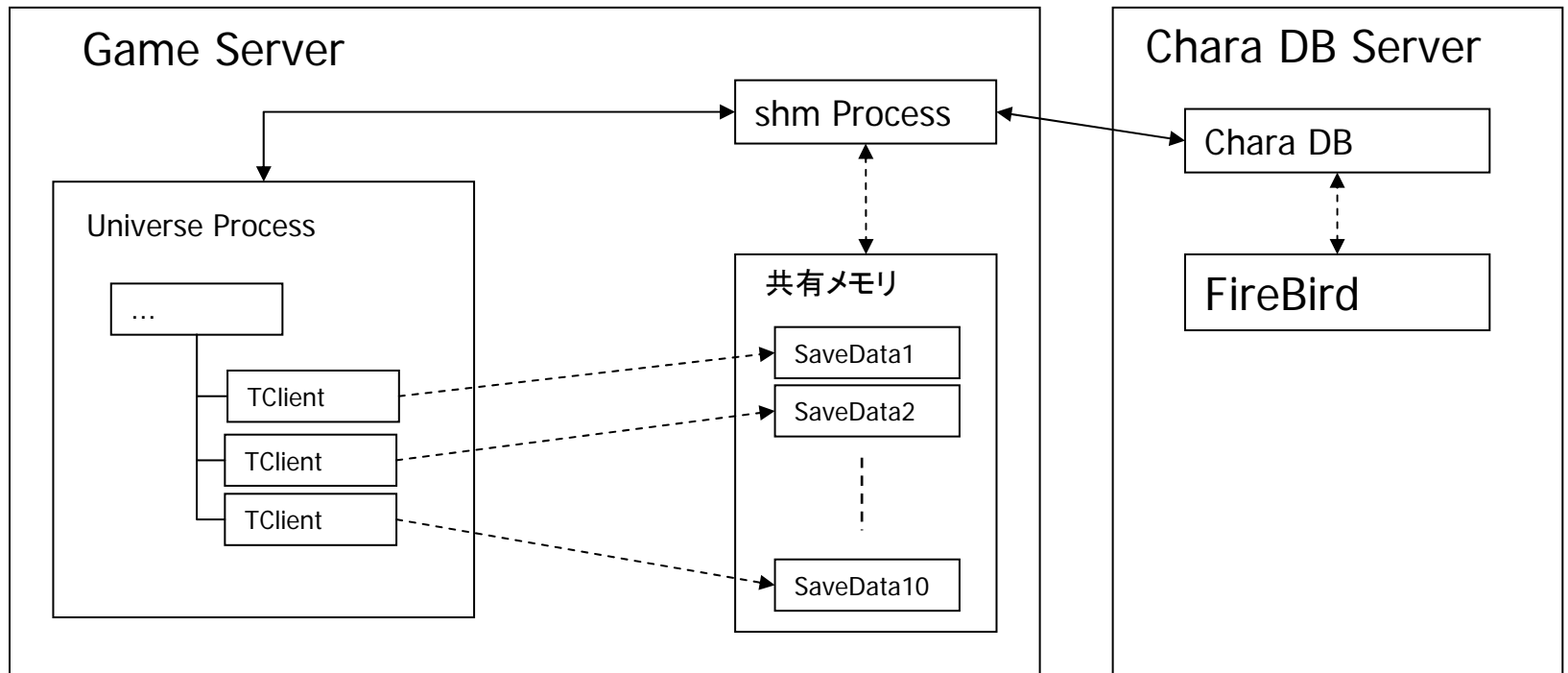
# パケット削減の仕組み



- MMOの為にパケット削減処理が必要
- X,Z軸座標10m単位のグリッドツリーを作成
- プレイヤーからの移動パケットが来る度にグリッドを移動する
- 自分のいるグリッド、その周りのグリッド、さらにもう一回りのグリッドでパケットの送信頻度を低下させる。1/1、1/4、1/16、といった感じ。



# 共有メモリDBイメージ図



# 共有メモリDBの仕組み

- DBサーバーへの負荷が掛からないように、共有メモリ上にDBを作成する。
- クライアントが生成された際、共有メモリDBへデータのリクエストをする。
- DBサーバーからデータを読み込み共有メモリにセットし、データ構造体のポインターを返す。
- プレイヤーのパラメーターの変化があった場合、直接構造体のデータを書き換える。
- ゲーム終了、あるいは回線切断時にクライアントがデストラクトされる。その際共有メモリのデータがDBサーバーへ送られ、保存される
- プログラムがハングアップしても共有メモリDBのデータは消えない。
- もしプロセスが正常終了されなかった場合、次回起動時に自動的に保存処理を行う。
- サーバマシンが物理的に壊れた場合はどうしようも無いが、今の所問題は起きていない。

# チート対策について

- 次世代機になってネットワーク対応ゲームが増えてくるが、チートへの対策に関してはよく解っていない人が多いと思われる。
- 基本的に完全クライアントサーバー型にして処理を全てサーバー側で行わない限りチート回避は不可能。
- それでもクライアントでゲームの処理する場合、チートをどの程度まで許容するかゲームデザインでのトレードオフが必要になる。

# チートの手法

## ◎プログラム書き換え

- アイテムを使っても減らない
- 壁抜け、無敵
- 攻撃力増加

## ◎異常な値を利用する

- 逆レスタ
- 超巨大化、超マイクロ化

## ◎シーケンスの状態を利用する

- ロビーテクニック
- ログインシーケンス抜け
- ゲーム乱入時のパケット細工

## ◎タイミングの間隙を利用する

- アイテム交換、預かり所でのデュープ
- 予期しないタイミング系のセキュリティーホールは非常に発生しやすい

## ◎バッファオーバーを利用したもの

- チャットの文字列あふれ
- メール文字列あふれ
- パラメーターは必ずチェックする
- strcmp、strcat、strcpy等をそのまま使わない

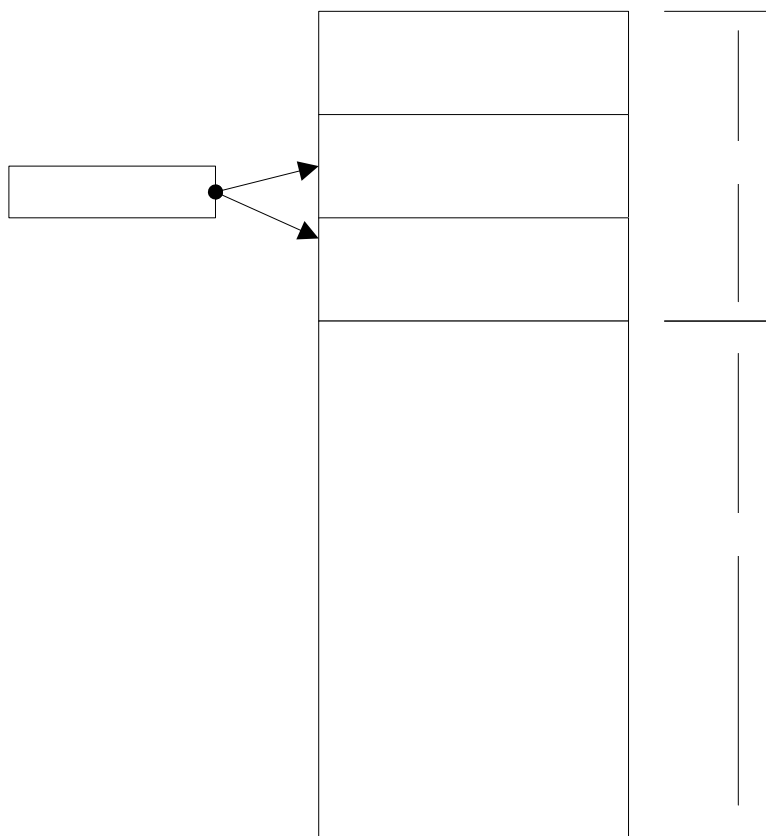
# プログラムする上で気を付ける事

- クライアント、サーバー共に受信したパケットは受信側で完全にチェックする。  
(どんな値が来てもハングアップしないようにする。)
- 状態に依存するようなプログラムを書かない
- シーケンスに依存するパケットは受信関数テーブルの切り替え等で対処
- 異常パケットが来たらログを残して即切断する
- 重要な部分は全てサーバー側で処理する

# コンシューマーゲームのチートに関して

- コンシューマーゲーム機にはプロアクションリプレイ等のチートツールが出回っている。
- 法律的にはゲームの改造自体は違法ではない。
- ネットワークゲームの場合は営業妨害として訴える事が出来るのかどうかは微妙。
- 法的な整備が望まれる。

# プロアクションリプレイの仕組み



- プロアクションリプレイのCDを起動するとシステム領域の隙間メモリにチートプログラムが常駐される。
- ゲームDISKを入れると自動的に再起動する。
- タイマー割り込みなど色々な条件でプログラム発動
- ゲームの該当箇所プログラムが書き換えられる

Memor

# プロアクションリプレイへの対策

- 割り込み等のベクターが変更されていないか調べる
- プログラム開始時、システム領域の隙間やスタックエリアのメモリをクリアする
- チートプログラムを発見したら、密かにサーバーにチート検出した事を知らせる
  
- 当然ハードの詳細を知らないといけない
- 将来的にBIOSのバージョンアップでアドレスが変わる事がある。最悪ゲームが起動しなくなる場合もある
- いずれにしてもPAR側で対策を打たれたら終わり



# PCでのチートに関して

- PCではチートの手法は非常に多岐に渡っている。
- Windows APIを使って簡単にチートツールが作成できる。
- ファイルも中身がまる見えで解析されやすい。
- Windowsは技術資料が多いのでハッカー人口も非常に多い。
- クライアント自体のパッチは当てやすいので、随時更新して行くしかない。

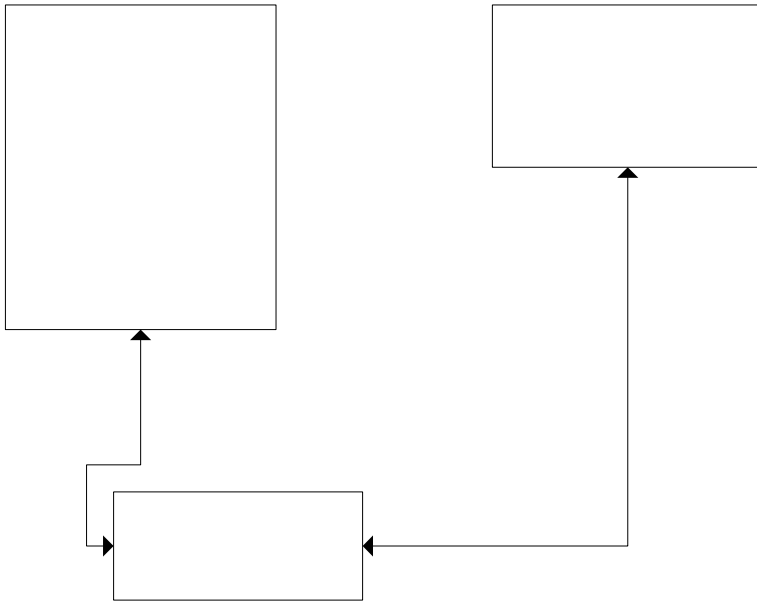
# PCチートその1 EXE書き換え

- EXEファイルの暗号化が行われていないと簡単にプログラムを書き換えられてしまう。
- 暗号化ツールも有名な物は殆ど対策されているので、フリーの暗号化ツール程度では気休めにしかない。
- 最終的にデバッガでメモリに展開されたイメージを吸い出されたら終わり。
- 自分で暗号化ツールを作るのはかなり大変。

# PCチートその2 ファイル書き換え

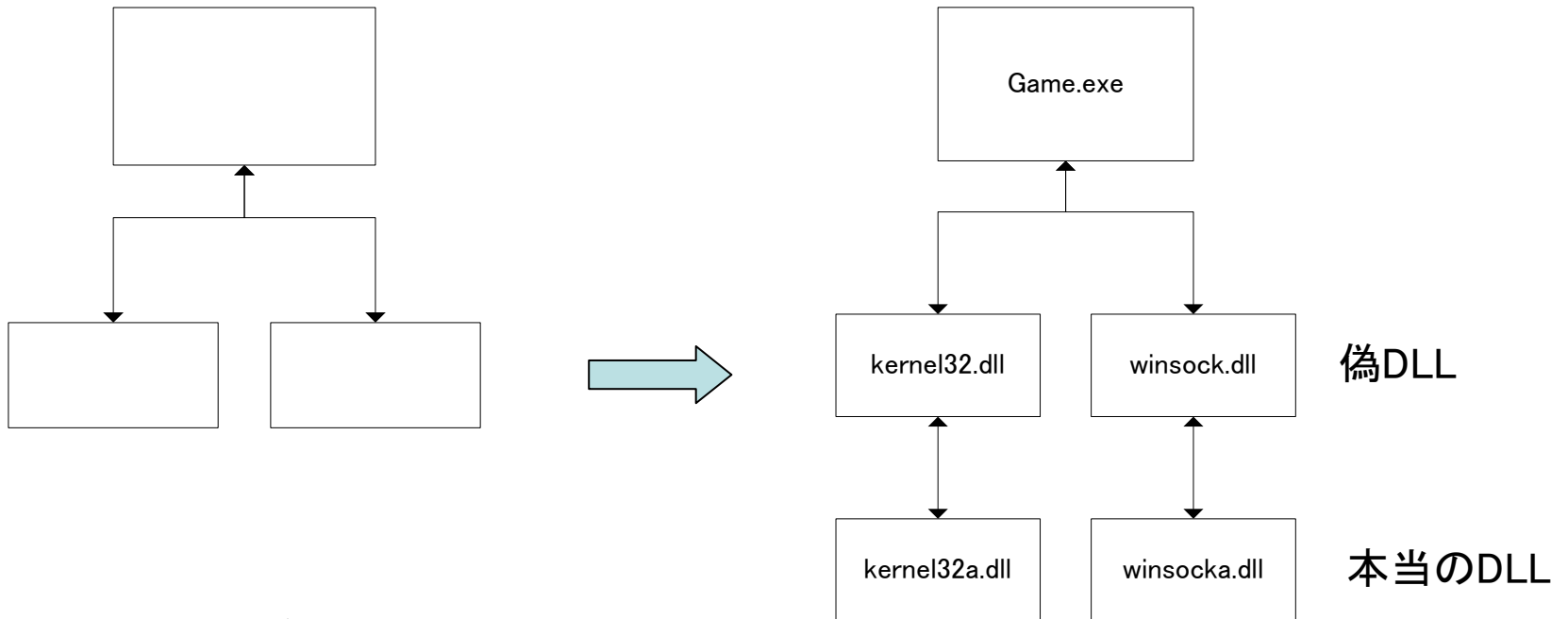
- EXEをいくら暗号化してもデータファイルの暗号化を忘れていると、そこを書き換えられる場合がある。
- マップのコリジョンデータを書き換えて壁抜けが出来たり、お店のアイテム情報を書き換えてレアアイテムを出したり、エネミーのデータを書き換えたり色々な事が出来る。
- ファイル名からどんなデータか類推出来たらダメ。ファイル名の暗号化も必要。
- 結局重要なデータはサーバーサイドに持たなければならない。

# PCチートその3 デバッグAPI



- Windowsにはデバッグ用のAPIがある。
- チートツールから他のプロセスにアタッチしてメモリの中身を見たり、書き換えたりする事が出来る。
- 殆どのチートツールはこれを使っている。
- Kernelレベルで動いているので通常では回避不能。
- これを防ぐにはWindowsのAPIを書き換えるような高度な技術が要求される。

# PCチートその4 偽装DLL



WindowsAPIは全てDLLの塊で出来ている。このDLLを差し替えて、Proxyの様に作用させる。

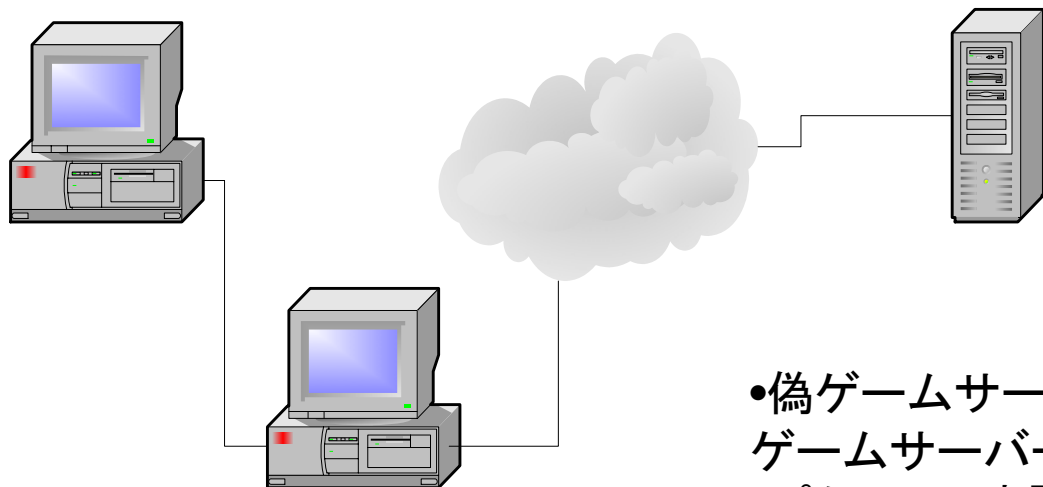
APIを自由に書き換えたり、データをキャプチャーしたり、何でも出来る。

Game.exe

# PCチートその5 高速加速器

- タイマー割り込み等をフックして、高速に回す手法。
- プログラムからは正常に動作しているように見えるが通常の倍速で動いていたたりする。
- プレイヤーの動きが速くなる事によって、他のプレイヤーを出し抜いて攻撃したり、タイムアタックでとんでもない記録を出したりする。

# PCチートその6 偽中継サーバー



- 偽ゲームサーバーを立ててそこを経由させてゲームサーバーへ接続する。
- パケットの暗号化ルーチンはexeから抜いて来る。
- 共通鍵を使用している暗号化ルーチンでは回避不能。
- パケットをフィルタリングして、要らないパケットを排除したり、不正なパケットを送ってサーバーを誤動作させたりする。

# PCチートその7 BOT

- 自動的にキー入力をしてくれるようなツールで作業を自動化する。
- 簡単なゲームだとロケットマウス等でマクロが組める。
- 中継サーバーを利用して、直接パケットを送信する物もある。パケットの暗号化が弱いと狙われやすい。
- 昔は3DゲームでBOTを作るのは難しいとされてきたが最近はそうでもないらしい。
- 各ゲームの専用ツールとして配布されるパターンが多い。
- BOTツールが起動しているかチェックするしか方法が無いが、いくら対策してもきりが無い。



# PCチートその8 スパイウェア

- ネットカフェなどを狙ってキーロガーを仕掛ける。
- 盗んだID、Passwordを利用してアイテムを盗んだり、チートの実験をしたり、リアルマネートレードに使ったりする。
- 最近ハードウェアタイプのキーロガーもあるらしい。
- 現状、ネットワークゲームで逮捕者が出ているのはID、Passを盗んだ場合。
- その他のチートは法律が整備されていないので訴えるのも難しい。

# PCチートその9 サーバー窃盗

- データセンターから直接ゲームサーバーを盗み出すという荒技。
- 日本ではありえない話だが、中国、韓国では結構頻発しているらしい。
- 運営会社の社員が密通している場合が多い。
- ドングルを使った暗号化やサーバー認証などサーバーサイドのプログラムにも何らかのセキュリティー対策が必要。

# PCチート対策ソフト

## ◎INCA社

### nProtect Game Guard

## ◎AnLab社

### HackShield

- デバッガAPIをふさいだり、チートツールを検出したり、EXEの暗号化をかけたり、スパイウェアを検出したり、考えられる様々なチートに対応。
- 自動更新なのでゲーム会社側は何もしなくてもいい。

## • nProtectの効果は？

- ・チートを完全に防ぐのは不可能。
- ・しかし本当にチートが出来る人はほんのわずか。
- ・カジュアルユーザーのチートの蔓延は防ぐという意味で非常に効果的。
- ・サーバー側のチート対策と合わせて運営で対応する必要がある。