CardSoft

セキュリティ ソリューション概要

2003 年 1 月

Zen Kishimoto, Ph.D.

zen.kishimoto@cardsoft.com

目次

1.0 はじめに

Java は Sun Microsystems によって開発され種々の環境に適用されてきた。Java の優位性が証明されるにしたがって、容量や性能が制限される組み込みシステム用 Java 環境である J2ME が定義された。この環境は用途とシステム容量に応じて、CDC（家電など）、CLDC（携帯電話や PDA）、および Java Card に分類されるが、このホワイト ペーパーでは CLDC について述べる。CLDC では、最低限必要なライブラリと Java のバーチャル マシン（VM）が定義されているが、KVM と呼ばれるこの小型 VM は、多くのメモリやストレージを必要としない。この KVM 上で Sun は 2 つのプロファイルを定義した。携帯電話用の MIDP、PDA 用の PDAP の 2 つである。図 1 にこれらの要素の関係を示す。

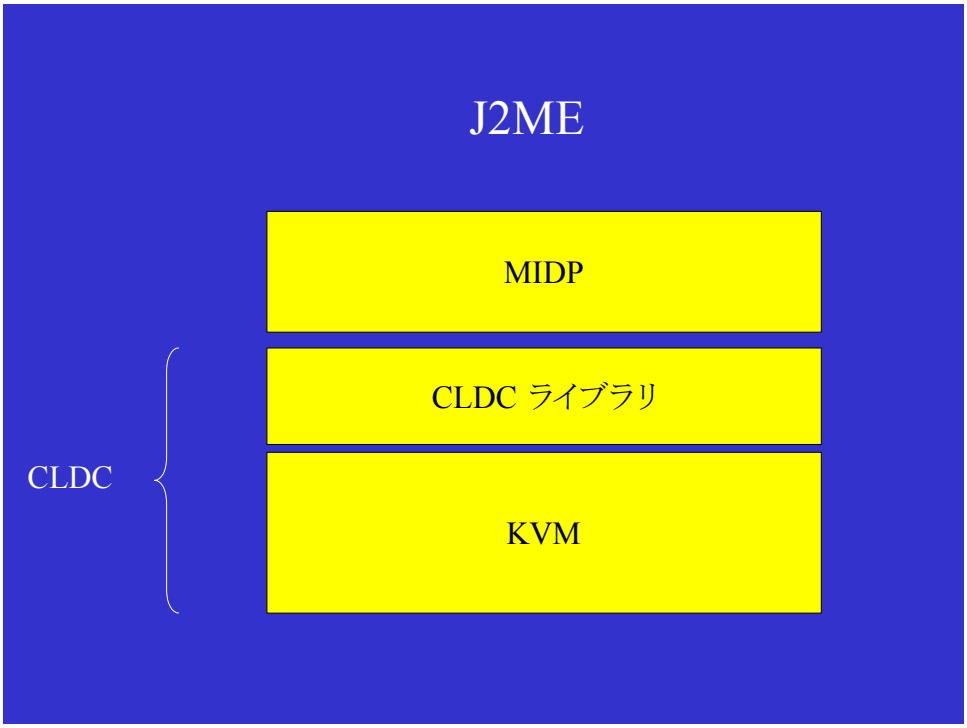

図 1　J2ME の要素

Sun が開発した KVM はスプリット VM を採用していないため、組み込みシステム用装置に対応しているとは言えず、また多くのメモリを要求し実行速度も遅いこともあって実用的ではない。さらに CLDC では、容量が小さい装置に Java を搭載する際のサイズ上の制限から種々の機能が削除され、セキュリティ機能も一部削除された。こうした問題を解決するのが、CardSoft 社[1]のソリューションである。

まず性能の問題について、CardSoft は J コンソーシアムのメンバーとしてスプリット VM を実

[1] www.cardsoft.com

現するファイル システムである JEFF[2]を共同開発した。このファイル システムを使用することにより、メモリ使用の 50%程度削減が達成できる。さらに CardSoft はセキュリティの問題に対応するため、他社と共に STIP[3]コンソーシアムを形成してセキュアなトランザクションを必要とするアプリケーション用の標準を設定し、それに基づいて製品を開発した。STIP では、サーバと装置間のセキュリティの他、装置内におけるアプリケーション間のセキュリティも考慮している。現在、STIP の最新版である 2.1 に完全準拠した製品を提供するのは CardSoft のみである。

MIDP

MIDP 1.0 はリファレンス実装であり、Java セキュリティの根幹をなす sandbox の機能が元の標準 Java に較べると格段に弱い。このため、金融機関が要求するセキュリティ レベルを満たしていない。MIDP 2.0 では、セキュリティは改善されてはいるが sandbox の強化には至っていない。TSG SA WG3 Security[4]（3GPP[5] system のセキュリティ委員会）のチェアである Mike Walker は、MIDP 2.0 の sandbox 機能は改善されていないのではないかと尋ねられ、以下のように回答してそれを認めている。

"That is correct, MIDP 2.0 did not improve the sandbox model as such. However, it contains a good security framework, which we can use to address some of our requirements. That is one of the reasons why we initiated 'The Recommended Security Policy for GSM/UMTS Compliant Devices' or the RP initiative. The RP document is an addendum in the MIDP 2.0 specification. Within this document detailed security policy is specified for GSM and UMTS handsets. This security policy is based on security domains with varying capabilities and privileges to give access control to sensitive capabilities on the terminal. In particular, attention is given to events that could be triggered by a MIDlet that could result in a charge to the end user."

MIDP 2.0 は、金融機関が要求するレベルのセキュリティを提供する基本レイヤとして設計されていない。セキュリティ機能は、アプリケーション レベルでのみ追加することは不可能であり、設計時に基本レイヤに組み込まれている必要がある。このため、MIDP 2.0 は、セキュリティを要するアプリケーション用の環境として不適当である。

STIP

STIP によるセキュリティは、金融トランザクション分野の標準化団体、GlobalPlatform[6]と FINREAD[7]（Financial Transactional IC Card Reader）により支持されている。GlobalPlatform は、スマートカードの振興を助成する団体であり、主要なクレジットカード会社が参加している。GlobalPlatform（GP）の仕様 2.0 では、デバイス API と STIP が密接に結合している。図 2 に GP と STIP の関係を示す。STIP 層上で GP を使用して Stiplet と呼ばれるアプリケーション開発が行われる。CLC（Chip Logic Component）はスマートカードとのインターフェース コンポーネ

[2] www.j-consortium.org/jeffwg/index.shtml
[3] www.stip.org
[4] www.3gpp.org/TB/SA/SA3/SA3.htm
[5] 3rd Generation Partnership Project
[6] www.globalplatform.org
[7] www.finread.com

ントで、Stiplet にリンクされる。また、EMV 仕様はスマートカードとそのリーダーのセキュアな通信を規定する。このように、GP および EMV 仕様で STIP 層は重要な役割を担っている。

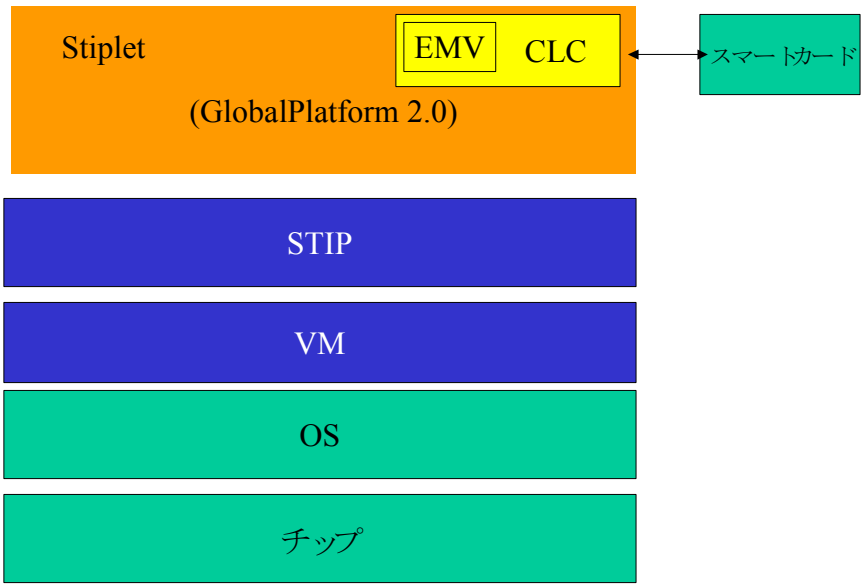

図 2　GlobalPlatorm と STIP

Visa と MasterCard を始めとするクレジットカード会社がセキュリティを強化する中、ヨーロッパでは EMV 仕様が 2005 年 1 月 1 日より必須となり、この仕様を実装しないクレジットカード発行者は不正による被害に対し救済が受けられなくなる。セキュリティを強化する動きは世界中に広がっており、例えば日本クレジットカード協会では 2003 年より磁気ストライプ カードに替えて IC カード使用を本格化させることを決定した。こうしたセキュリティ強化の要求により、小型装置を使用するコマース トランザクションにおける STIP の重要性はますます増加する。

GlobalPlatform とならび STIP を支持している FINREAD コンソーシアムはヨーロッパの金融機関のグループで、PC に接続したスマートカード リーダーからセキュアなトランザクションを可能にするための仕様を開発している。FINREAD による STIP のサポートでは、STIP API の一部が周辺機器のインターフェイスとして実現されている。INREAD の浸透に伴い、STIP の重要性も増加する。

2.0　セキュリティ

Java セキュリティの分類法は幾つかあるが、このホワイトペーパーでは sandbox、認証、暗号化に絞って議論する。この 3 点に関する Java、MIDP 1.0、MIDP 2.0、STIP、CardSoft STIP の比較を表 1 にまとめた。小型装置特有のセキュリティ問題で最も重要な要素は sandbox である。

これは、複数アプリケーションを搭載した装置を使用した金融トランザクションでアプリケーション間のセキュリティを確保するのが sandbox 機能であることによる。

		標準Java	CLDC/MIDP 1.0	CLDC/MIDP 2.0	STIP	CardSoft STIP
Sandbox	Java 言語特徴					
	良く定義されている	X	X	X	X	X
	自動ガーベッジ コレクション	X	X	X	X	X
	コンパイル時のチェック	X	X	X	X	X
	リフレクションの除去		X	X	X	X
	ダブルやフローティング タイプの除去		X	X	X	X
	アプリに対するスレッドの除去				イベントドリブン	イベントドリブン
	クラスローダ					
	デフォルトクラスローダ	X	X	X	X	X
	定義可能なクラスローダ	X				
	バイトコード検証					
	標準法	X				
	オフデバイス		X	X		
	独自法				X	X
	Security Manager					
		X			Access Manager	Access Manager
				セキュリティ ドメイン		セキュリティ ドメイン
	デジタル署名					
		拡張機能		X	X	X
	通信時の暗号化					
		拡張機能		X	X	X

表1 Javaセキュリティの比較

複数のアプリケーションを同時に実行する電子コマースで小型装置の使用が増加するにつれ、装置内に格納した複数のアプリケーション間で互いのデータを破壊したり不正にアクセスしたりする可能性が問題となってきた。これを防止する方法のひとつは、装置に複数のアプリケーションを搭載しないことであるが、これはアプリケーションごとに装置が必要となり現実的ではない。

MIDP や DoJa では、複数アプリケーションを同時に実行できないようにすることで、アプリケーション間で問題が起きるのを防止している。しかし、アプリケーションが他のアプリケーションのデータを改ざんしたり破壊することは起こり得るため、この方式も不完全であり、必ずしもセキュリティは保証されていない。以下に、sandbox その他のセキュリティ機能について述べる。

2.1 Sandbox のセキュリティ

sandbox は、各アプリケーションを論理的に囲い込む壁として機能し、アプリケーションのアクセスをあらかじめ許可されたリソースのみに制限する。sandbox は以下の 4 つの要素より成る。1) Java 言語の機能、2) バイトコード検証、3) クラスローダ、4) Security Manager。

2.1.1 Java 言語の機能

Java は強く型付けられた言語で、セキュリティ関連の特徴として、ポインタを使用しないこと、自動ガーベッジ コレクション機能があることが挙げられる。STIP ではこれに加え、アプリケーションに複数のスレッドを使用せずイベント ドリブンのアプローチを採ることにより、デッドロック等の問題を防いでいる。

2.1.2 バイトコード検証

標準 Java ではバイトコードの検証は VM で実行時に行われるが、CLDC ではオフ デバイスで行われる。このため VM の小型化が達成されたが、セキュリティに関する影響はあまりないと言えよう。STIP ではバイトコード検証を実装時に行うようにしており、CardSoft では独自の方式を使用している。

2.1.3 クラスローダ

標準 Java と異なり、CLDC と STIP では新たにクラスローダを定義することは禁じられており、デフォルト クラスローダのみが使用される。アプリケーションが専用のクラスローダを定義できないので、セキュリティ ホールが生じることがない。この機能はネットワークで結ばれた環境で重要である。

2.1.4 Security Manager

Security Manager は標準 Java において sandbox の輪郭を規定するものである。小型装置の環境ではストレージ容量に制限があるため、MIDP 1.0 と CLDC 1.0 では Security Manager が削除され、その結果 sandbox が弱体化された。MIDP 2.0 ではセキュリティ ドメインの概念が導入されたが、Security Manager に代わる機能は取り入れらていない。

これに対して STIP では Security Manager に代わる機能、Access Manager を定義した。Access Manager を介してデバイス コントロールを行い、これをセキュリティ ドメインと組み合わせることにより、アプリケーションがデバイス リソースに不正アクセスすることを防ぐ。ファイルや周辺機器などのデバイス リソースはすべて、デバイス コントロールを介してのみアクセス可能であり、デバイス コントロールの作成はアプリケーションによって行うことはできず、Access Manager を通す必要がある。Access Manager は、アプリケーションに対し設定されたセキュリティ ドメインによって、デバイス コントロールを介してアプリケーションにアクセスの権利を与えるか、または拒否する。このメカニズムにより、周辺機器への不正なアクセスが防止され、sandbox の境界が保護される。

CardSoft は Access Manager と緊密に動作する独自のセキュリティ ドメインを構築した。MIDP

2.0 では API と機能へのアクセス コントロールに対しセキュリティ ドメインを設定しているが、CardSoft STIP ではすべてのリソースに対する ACL（アクセス コントロール リスト）を定義しており、MIDP 2.0 で定義されたものより総合的である。

2.2 認証

認証は、アプリケーションが実際に信頼できるサイトから来ていることを確認する手段である。信頼できるサイトからのものであれば、実行時に機密を要する情報へのアクセスが与えられる。認証は通常、保証書を交換することで実現する。

MIDP 2.0、STIP の両者ともアプリケーションの認証を行う。実際にどのように認証を実現するかはそれぞれのベンダーに負う所が大きいが、STIP では Access Manager を通して周辺機器へのアクセスをより細かくコントロールすることが可能で、セキュリティのポリシーを設定する上で柔軟性がある。

2.3 サーバとの通信におけるセキュリティ

サーバとの通信におけるセキュリティは暗号化によって実現される。MIDP 2.0/CLDC および STIP では暗号化の API が定義されており、ベンダーが提供した暗号化の手法に対するラッパーを提供する。

3.0 MIDP と STIP の統合

MIDP は携帯電話ですでに広く使用されているが、金融トランザクションに使用するにはセキュリティが不十分である。しかし、STIP を搭載したセキュアなトランザクション用携帯電話を、MIDP を搭載した携帯電話と別途所有するのは実用的ではない。こうした点を考慮すると、すでに多くのベンダーからサポートされアプリケーションも豊富な MIDP と、金融機関からサポートされている STIP を統合できれば理想的である。MIDP と STIP のアプリケーションは完全に独立して開発されてきたが、どちらのアプリケーションも変更せずに同じプラットフォーム上で実行可能である。

MIDP と STIP の統合には次の 2 つのオプションがある。

- オプション 1: STIP のライブラリを MIDP のプラットフォームに移植する。
- オプション 2: MIDP のライブラリを STIP のプラットフォームに移植する。

どちらのオプションでも、VM の変更は必要であるがアプリケーションの変更は必要ない。これは Java の「1 度書けばどこでも実行できる」というスピリットから見て大変重要である。VM への変更程度は 2 つのオプションで異なる。図 4 に示すオプション 2 は、図 3 に示すオプション 1 より VM に多大な変更を必要とするが、JEFF ファイル システムによりメモリを効率的に使用でき、そのため実行速度も速くなる。長期的にはオプション 2 が適切である。なお、STIP 環境では JEFF ファイルを使用することが望ましい。

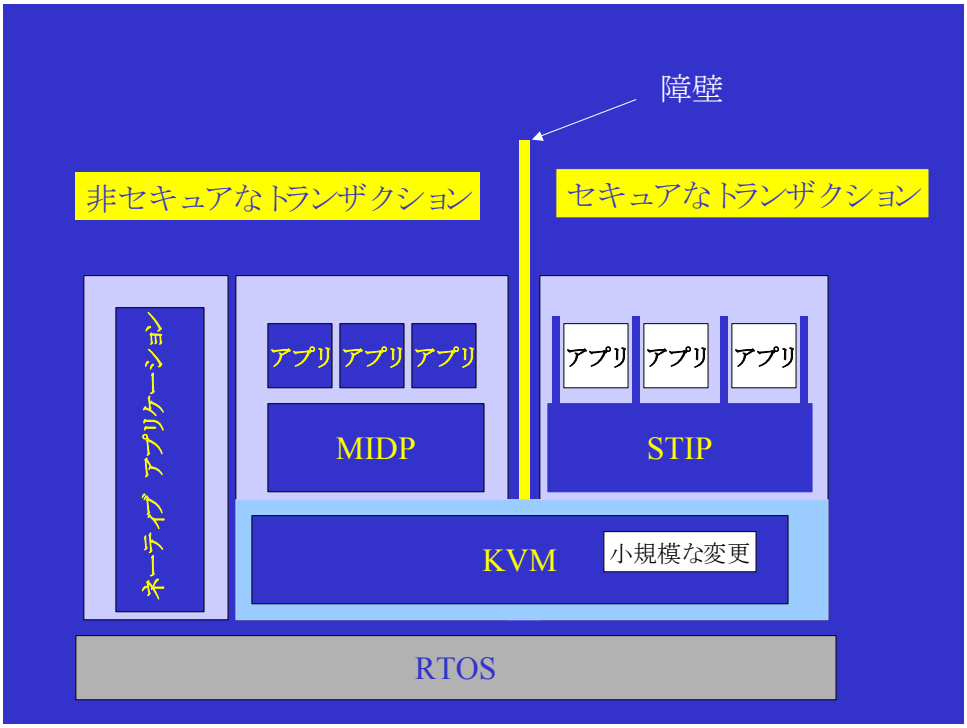

図 3　オプション 1

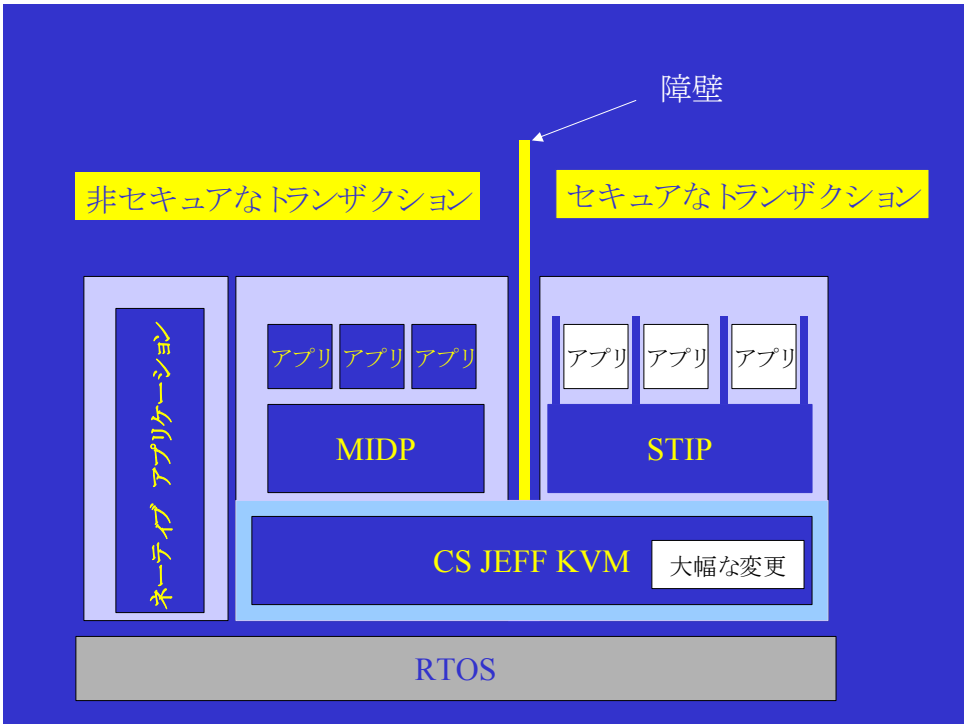

図 4　オプション 2

4.0 CardSoft の製品

CardSoft では eAppliance™ソリューション スィートを提供している。

CardSoft JEFF VM は RAM に対し最高のスピードを提供。Java VM の「クリーン ルーム」実装であり、Sun の Java 1.02 仕様に準拠。さらに最近 ISO 標準に認定された Java 実行ファイル形式である JEFF（J Consortium による）を実装しており、RAM の使用量を大幅に削減できる。

eASP v1.0（eAppliance STIP プラットフォーム）は Java 実行環境で、STIP アプリケーションを搭載、実行可能。現在、最新の STIP 2.1 を出荷しているのは CardSoft のみである。CardSoft の STIP は、標準 STIP 機能だけでなく、他の STIP ベンダ製品にはない CardSoft 独自の機能も付加されている。

eAFP v1.0（eAppliance FINREAD プラットフォーム）はインテリジェント スマートカード リーダー用の FINREAD 仕様を実装。FINREAD は STIP を採用し、CEN（ヨーロッパ標準化協会）を通じて電子コマース、ホーム バンキングや電子署名用のインテリジェント スマートカード リーダーの仕様を開発している。FINREAD のスマートカード リーダー キャンペーンは、オープンなネットワーク上のセキュア バンキング世界標準へ繋がると予想される。

eADE（eAppliance 開発環境）は、STIP や FINREAD のアプリケーションを迅速に開発するためのツール セット。これには、Borland 社の Jbuilder™プラグイン、コマンド ラインの JEFF コンバータ、PC ベースの eASP 実装が含まれる。

5.0 結論

標準 Java では、エラーを含んだコードや有害なコードから実行プラットフォームを保護する方策が講じられている。しかし容量の小さい装置では、これを実行することは不可能である。
STIP はセキュアなランタイムを保証する Java 環境であり、スマートカードやクレジットカードなどの使用で金融機関が求めるセキュリティを保証するために必須である。MIDP は J2ME のリファレンス技術として実装されその基本設計にセキュリティが組み込まれていないが、STIP は当初から銀行やクレジットカード会社が要求するセキュリティを満たすように設計され実装されている。このため STIP は FINREAD と GlobalPlatform により、基本アーキテクチャとして選択された。その上 JEFF ファイル形式を採用しているため、STIP のアプリケーションは MIDP に比べ高速でメモリの消費も少ない。

STIP はセキュリティ ドメインを使用することにより、セキュリティを強化した。MIDP 2.0 にも同様の機能があるが、STIP ではデバイス コントロールと Access Manager の概念を導入することにより、総合的でなおかつきめ細かいコントロールを実現できる。

CardSoft では、MIDP と STIP を、それぞれ既に開発されたアプリケーションを変更することなく統合する方法を検討している。携帯電話や PDA のような小容量の装置が頻繁に使用されるようになると、厳しいセキュリティを必要とするアプリケーションとそうでないアプリケーションが混在するようになる。MIDP と STIP の統合により、1 台でどちらの要求にも応えることが可能となる。