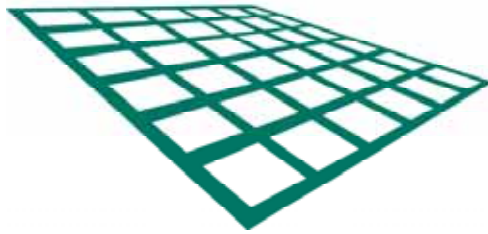


Enterprise
Grid Alliance



エンタープライズ・グリッド・
セキュリティ要件

策定:

エンタープライズ・グリッド・アライアンス
セキュリティ・ワーキンググループ

*** Version 1.0 ***

EGA 理事会承認

2005年7月8日

Enterprise Grid Security Requirements v1.0 日本語訳

2005 年 8 月

この文書は、『Enterprise Grid Security Requirements v1.0』文書の翻訳である。この翻訳文書と原書である英語文書との間に食い違いがある場合、又は、翻訳文書に省略がある場合、原書の英語文書が決定版として扱われるべきである。

日本語訳作成

エンタープライズ・グリッド・アライアンス日本運営委員会

日本語訳貢献者

貢献者	所属
阿部 恵史	日本ヒューレット・パカード株式会社
加藤 雅之	日本電気株式会社
倉橋 秀則	日本ネットワーク・アプライアンス株式会社
古城 隆	日本電気株式会社
後藤 哲也	EMC ジャパン株式会社
末廣 謙二	日本電気株式会社
鈴木 俊宏	日本オラクル株式会社
蒔田 賢治	日本オラクル株式会社

著作権表示

Copyright ©2005 Enterprise Grid Alliance. All rights reserved. 本書に含まれる情報は、エンタープライズ・グリッド・アライアンス（EGA）の書面による事前の許可なしに、出版、配布、訂正、再配布することはできません。本書の使用許諾についてのお問い合わせは、エグゼクティブ・ディレクタ（EGA_Executive_Director@gridalliance.org）までお送りください。

Author Information

<i>Contributor</i>	<i>Organization</i>
Mike Beckerle	Ascential Software Corporation
Glenn Brunette	Sun Microsystems, Inc.
Lee Cooper	Oracle Corporation
Wan-yen Hsu	Hewlett-Packard Company
Adam Jacobs	Oracle Corporation
Thomas Keefe	Oracle Corporation
Richard Nicholson	Paremus, Ltd.
Parviz Peiravi	Intel Corporation
Collin Sampson	Sun Microsystems, Inc.
Bob Thome	Oracle Corporation

改訂履歴

バージョン	状態	日付	コメント
1.0		2005年7月8日	一般公開用の初版 v1.0 ドキュメント

目次

1	はじめに	1
1.1	エンタープライズ・グリッド・アライアンス	1
1.2	グリッド・セキュリティ・ワーキンググループの目的	2
2	エンタープライズ・セキュリティの基礎	4
3	EGA参照モデルのセキュリティ	5
3.1	グリッド・コンポーネントのセキュリティ	5
3.1.1	グリッド・コンポーネントのライフサイクル	8
3.1.1.1	プロビジョニング	9
3.1.1.2	継続的管理	10
3.1.1.3	デコミッションングおよび目的の再割り当て	11
3.2	グリッド管理エンティティのセキュリティ	12
4	グリッド・セキュリティのユースケース	15
4.1	汎用ユースケース－グリッド・コンポーネントのプロビジョニング	15
4.1.1	グリッド・コンポーネントの追加/作成	15
4.1.2	グリッド・コンポーネントの構成	16
4.1.3	グリッド・コンポーネントの開始	16
4.2	汎用ユースケース－グリッド・コンポーネントのモニターと管理	17
4.3	汎用ユースケース－グリッド・コンポーネントのデコミッションング	18
4.3.1	グリッド・コンポーネントの停止	18
4.3.2	グリッド・コンポーネントの構成解除	18
4.3.3	グリッド・コンポーネントの削除	19
4.4	運用ユースケース	19
4.4.1	ユーザー/サービスのセキュリティ・イベント	19
4.4.2	ワークロードのセキュリティ・イベント	19
5	グリッドの脅威とリスク	21
6	グリッド・セキュリティ要件	24
6.1	機密性 (Confidentiality)、完全性 (Integrity)、可用性 (Availability) - (CIA)	24
6.2	識別 (Identification)	25
6.3	認証 (Authentication)、認可 (Authorization)、監査 (Auditing) - (AAA)	25
6.4	任務の分離、最小の特権	25
6.5	縦深防御 (Defense in Depth)	26
6.6	フェイル・セキユア	26
6.7	グリッド・ライフサイクルのセキュリティ要件	26
6.8	相互運用可能なセキュリティ	27
6.9	安全な分離	27
6.10	信頼関係	28
7	要約	29
8	関連活動	30

8.1	標準関連活動.....	30
9	参照情報.....	31



図 1 - 抽象コンポーネントから具体コンポーネントへのマッピング 1	5
図 2 - 抽象コンポーネントから具体コンポーネントへのマッピング 2	6
図 3 - 一般的なグリッド・コンポーネントのライフサイクル	9
図 4 - グリッド・コンポーネントと論理的に区別されるGME	13
図 5 - データ・センターの基本的概念	21

前書き

この資料は、エンド・ユーザー企業、関連する標準化団体、ベンダー向けに、エンタープライズ・グリッド・アーキテクチャに固有のセキュリティ要件を示すために作成された。この資料は、「エンタープライズ・グリッド・アライアンス（「EGA」）参照モデルv1.0」で定義された、参照モデル、ユースケース、ライフサイクルをベースとしているため、読者は、この資料を読む前に、まず「エンタープライズ・グリッド・アライアンス参照モデルv1.0」を読まれるよう強くお勧めしたい。

本資料は、以下のセクションから構成される。

はじめに

エンタープライズ・セキュリティの基礎

EGA 参照モデルのセキュリティ

グリッド・セキュリティのユースケース

グリッド・セキュリティの脅威とリスク

グリッド・セキュリティ要件

内容と形式についてコメントをお寄せいただければ幸いです。宛先は、EGA グリッド・セキュリティ・ワーキンググループ（ega_gridsecuritywg@mail.gridalliance.org）です。

(空白ページ)

1 はじめに

1.1 エンタープライズ・グリッド・アライアンス

エンタープライズ・グリッド・アライアンス (EGA) は、グリッド・コンピューティングの採用促進、またエンタープライズ・データ・センターにおけるグリッド・コンピューティングのデプロイメントと使用を実現するための技術の採用促進を目的としたコンソーシアムである。EGA は、エンタープライズ・グリッド・コンピューティングで実現すべき長期的な目標を見据えつつも、現実的なアプローチとして、当面はエンタープライズ・グリッド・コンピューティングの採用を加速するための短期的、中期的な目標に焦点を当てている。

グリッド・コンピューティングの特徴は、ネットワーク上に分散したリソースのプールを共有、管理してアプリケーションやサービスを提供するというものである。グリッド・コンピューティング環境には以下のような特徴もある。

- 個々のリソースをネットワーク分散型の共有可能プールを使用して管理し、優れた戦略的敏しょう性、アーキテクチャの柔軟性、パフォーマンス、スケーリング、耐障害性、可用性を実現する。
- 特に、エンタープライズ・グリッドによってネットワークは任意に構成されるリッチかつ複雑なリソース構造体として捉えられることになるため、個々のコンポーネントの管理ではなく、サービスの管理に焦点が当てられる。
- 変化する目標、規則、ビジネス目標に合わせて、あるいは単にサービスのニーズに合わせて、定期的にサービス・コンポーネントを構成したりサービス・コンポーネントの用途やプロビジョニングを変更したりすることが行える柔軟性または可変性。
- 分割あるいは分散される性質を持ち、リソース構造体の特性を活用するアプリケーションまたはサービスのアーキテクチャ。たとえば、ERP や CRM などの従来型の複数層アプリケーション、サービス指向アーキテクチャ (SOA)、分散可能な計算主体の処理など。
- コンピューティング・コンポーネント群を (通常は少数の) 大きなリソース・プールに統合することによってプロビジョニングを容易にし、サービスの可用性を高め、リソースの使用効率を上げ、管理を単純化する。
- 各種コンポーネントとそのインタフェース、構成、プロセス、アプリケーションを標準化することによって、変化するビジネス要件やサービス要件に効率的に対応できる、高度に自動化された耐障害性のあるアーキテクチャを推進する。

エンタープライズ・グリッド・コンピューティングとは、グリッド・コンピューティングを学術 / 研究目的ではなく、特に企業 (エンタープライズ) の枠組みの中で使用することを表す言葉である。エンタープライズ・グリッド・アーキテクチャにおける要件と他の種類のグリッド・アーキテクチャにおける要件の中には重複するものがあるかもしれないが、企業でのグリッド・アーキテクチャの採用に関しては、特に運用面で、固有の要件と課題がある。

エンタープライズ・グリッド・アーキテクチャは通常、単一の企業や事業体によって管理される。その単一の組織は、ネットワーク化された共有可能なリソース・プールを作成し管理すること、個々のリソースから高位のコンポーネントとサービスを構成すること、そして、定義された一連の目標と要件を満たすだけでなく、ビジネスの価値を高めることにも役立つサービスを提供することに関して責任を持つ。リソースは、コンピュータ、ネットワーク、ストレージのほか、サービス機能という形を取ることもある。すべてのリソースとサービスが、ある単一の企業に所有される場合もあれば、そうでない場合もある。組織は、ポリシーとビジネス目標に従って、サービス・プロバイダや、管理対象となるサービスを行うアウトソーシング会社などのような別の事業体のリソースとサービスを活用することもある。つまり、エンタープライズ・グリッドの境界は、管理上の責任と制御権の範囲によって定義されることになる。エンタープライズ・グリッドは、1つのデータ・センター内に収まる場合もあれば、複数のデータ・センターにまたがる場合もある。エンタープライズ・グリッド・アーキテクチャのサイズと適用範囲には、通常地理的な制限はない。ただし、リソースとサービスの実際の物理的な場所がどこであろうと、エンタープライズ・グリッド・アーキテクチャで管理されるリソースとサービスは通常、単一組織の管理責任と制御の配下に置かれる。

EGA について、また、EGA ワーキンググループそれぞれの適用範囲については、公開資料の「Accelerating the Adoption of Grid Solutions in the Enterprise（企業におけるグリッド・ソリューションの採用の加速）」および「Enterprise Grid Alliance Reference Model v1.0（エンタープライズ・グリッド・アライアンス参照モデル v1.0）」でより詳しく説明されている。これらはいずれも EGA Web サイト（<http://www.gridalliance.org>）から入手できる。

1.2 グリッド・セキュリティ・ワーキンググループの目的

EGA のグリッド・セキュリティ・ワーキンググループ (EGA-GSWG) の目的は、エンタープライズ・グリッドのアーキテクチャとコンピューティングに固有なセキュリティ上の脅威、課題、要件を特定し、（可能であれば）既存の技術や新しい技術を使用してこれらの要件を満たす方法、プロセス、そして推奨実現例を示すことにある。コンピュータ・システムとストレージがネットワークに接続され、人的リソースによって管理される従来のデータ・センターと同じように、エンタープライズ・グリッド環境における情報セキュリティの管理は、リスク管理の実践であり、そこから得られる益はリスクを補ってなお余りある。EGA-GSWG は、ユーザーがリスクにさらされることを制限しつつ価値を引き出せるように、エンタープライズ・グリッドのリスクと情報セキュリティの検討を公に行っている。

EGAのすべてのワーキンググループでは、単一のデータ・センターにおける商用エンタープライズ・アプリケーションに最初の焦点を当てて活動している。商用エンタープライズ・アプリケーションは、ほとんどの組織にとってその生命維持装置とも言えるものであり、顧客、パートナー、従業員、株主へのコンテンツとサービスの提供や、サプライ・チェーンやビジネス運営の編成と管理に関係している。これらのサービスは複数層で構成されることが多いが、そのようなアーキテクチャ構成は必須要件ではない。また、商用エンタープライズ・アプリケーションはバッチ式と対話式の両方のコンポーネントを持ったり、地理的に分散されたり、商用またはオープン・ソースのソフトウェア・パッケージの上に構築されたり、さらには組織のニーズに合わせてかなりの程度カスタマイズされることも多い。こうしたアプリケーションの例として、CRM、ERP、BIなどがあるが、それらのアプリケーションには、デフォルトでエンタープライズ・グリッドに対応しているものと対応していないものがある。エンタープライズ・グリッドに対応していないアプリケーションでも、コネクタやプロキシなどの仕組みを利用することによって、エンタープライズ・グリッド・アーキテクチャに組み入れることが可能である。

EGAは将来的に、ワーキンググループの適用範囲を複数データ・センターのモデルや技術エンタープライズ・アプリケーションにも広げる予定である。技術エンタープライズ・アプリケーションでは、計算主体の傾向がさらに強まり、対話処理は少なくなる。

EGA-GSWG グループの適用範囲には、コンポーネントが一元管理されており、それらが共有されたり用途が随時変化したりするエンタープライズ・グリッド環境に固有なセキュリティ上の課題が含まれる。このワーキンググループの目的は、（非グリッドつまり従来型の）エンタープライズ・セキュリティの管理やベスト・プラクティスに関連した議論をやり直したり蒸し返したりすることにはない。エンタープライズ・グリッドに固有のセキュリティの課題を認識することによって、組織は、自分たちの環境にエンタープライズ・グリッド・コンピューティングを採用する際、リスク管理に関わる適切な意思決定を行うための必要な情報を得て、十分な備えをすることができる。さらにベンダーも、製品や技術の機能拡張にこれらの情報を活用することで、それらをより競争力のあるものにすることができるだけでなく、セキュリティに関する顧客ニーズへの対応がより容易になる。

バージョン1となる本資料では、エンタープライズ・グリッド・セキュリティ要件に焦点を当てている。本資料の今後のバージョンでは、既存の技術を使用してこれらの要件を満たす方法を取り上げる予定である。さらにこのワーキンググループやEGAの他のワーキンググループがそれぞれの適用範囲を広げて単一データ・センターのユースケースの枠から出るときには、フェデレーション、組織間の信頼モデル、連携管理と連携モニタリングの手法といったセキュリティの論題も必然的に取り上げられることになる。

2 エンタープライズ・セキュリティの基礎

エンタープライズ・グリッド環境がエンタープライズ・コンピューティングの発展的形態であることを考えると、エンタープライズ・コンピューティングに影響するセキュリティ・ポリシー、要件、規則が、当然この新しい環境にも当てはまるということを確認する必要がある。同様に、エンタープライズ・グリッド・コンピューティングはエンタープライズ・コンピューティングから一歩進んだ論理モデルでもあるので、多くの要件、アーキテクチャ・パターン、推奨手法がエンタープライズ・グリッド環境にも適用されることを見て取れる。

エンタープライズ・セキュリティの論題は、学術分野、産業界の企業、コンソーシアム、標準化団体、および政府機関によって概ね十分にカバーされている。これらのグループは、主として以下の領域に関連するセキュリティ制御に焦点を当ててきた。

- プラットフォームとアプリケーションのセキュリティの極小化と強化
- プラットフォームとアプリケーションの認証、アクセス制御、監査
- ネットワーク・セキュリティに関する構成、フィルタリング、モニタリング、暗号化

これらの領域の成果の多くはインターネット上で広く公開されており、ベンダーやインターネット・セキュリティ・センター（CIS：Center for Internet Security）などのコンソーシアムの Web サイト、NIST や NSA が提供している政府機関の Web サイトなどで自由に入手できる。それらの情報は、理論や一般的手法から特定の製品構成や運用上の推奨事項まで、あらゆる点をカバーしている。これらの論題は特に新しいものでなく一般によく理解されて取り上げられていることを踏まえ、このワーキンググループはこれらの組織によって検討された成果を、エンタープライズ・グリッド固有のセキュリティ推奨仕様を策定する際の基礎として活用することにした。

3 EGA 参照モデルのセキュリティ

EGA 参照モデルでは、エンタープライズ・グリッドは次のように定義されている。

エンタープライズ・グリッドは、**グリッド管理エンティティ**の管理下にある、相互接続（ネットワーク化）された**グリッド・コンポーネント**の集合である。

また、エンタープライズ・グリッドと従来の典型的なデータ・センターの主な違いは、次のことを可能にする管理手法および技術にある。

コンポーネント中心の管理ではなく、サービスやアプリケーション中心の管理。

ネットワーク化されたリソースのプール化と共有。

以下のセクションでは、EGA 参照モデルに基づき、エンタープライズ・グリッド・アーキテクチャのセキュリティに特化した様々な側面を詳述する。EGA 参照モデルについて詳しくは、「エンタープライズ・グリッド・アライアンス参照モデルv1.0」を参照されたい。

3.1 グリッド・コンポーネントのセキュリティ

グリッド・コンポーネントは、エンタープライズ・グリッド内で管理されるすべてのコンポーネントの派生元となるオブジェクトのスーパークラスと定義される。これには、サーバー、ネットワーク・コンポーネント、ディスク・アレイから、データベース、ERP サービス、オンライン・ブックストアなどのようなアプリケーションやサービスまで、あらゆるものが含まれる。グリッド・コンポーネントには、様々な組み合わせることによって一般により高機能の要素（それ自体もグリッド・コンポーネントとなる）を形成できるという性質がある。

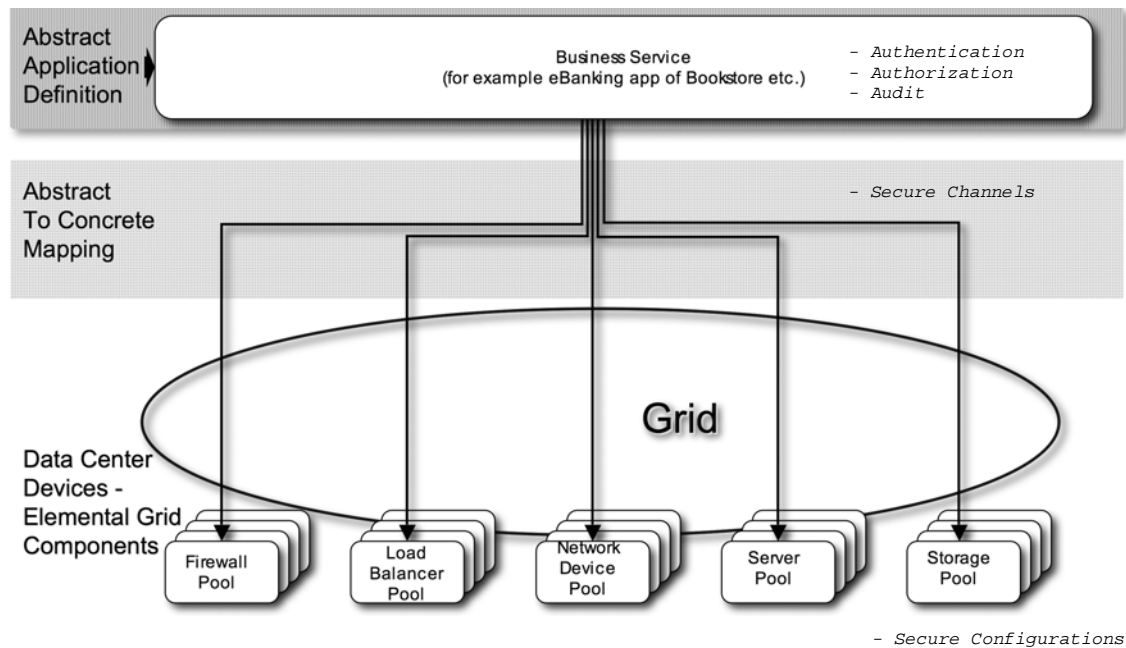


図 1 - 抽象コンポーネントから具体コンポーネントへのマッピング 1

図1は、サービスとその下層にある物理/論理グリッド・コンポーネント間の抽象化レベルの高いマッピングの簡単な例である。物理/論理グリッド・コンポーネントのそれぞれは、独自のセキュリティのプロパティと制御（つまり構成）を持つことを理解しておく必要がある。これは、事実上、現在の個々の製品やコンポーネントのデプロイメント方法と何ら変わりはない。必要なレベルのセキュリティと保証を達成するため、個々のコンポーネントはそれぞれが安全なものであり、また、安全な仕方での他のコンポーネントと組み合わせられていなければならない。同様に、グリッド・コンポーネントの集合体や組み合わせは、それを構成する要素自体のプロパティの論理和に加え、集合体としての固有のセキュリティ要件やプロパティを持つ場合がある。これもまた、従来型のデータ・センターにおいて現在行われているコンポーネントのデプロイメント方法と何ら変わりはない。たとえば、インターネット・バンキング・アプリケーションにおける認可されたトランザクションの概念は、そのアプリケーションがエンタープライズ・グリッド環境にデプロイメントされていてもいなくても、様々な物理および論理コンポーネントへのマッピングに至る過程で保持される。エンタープライズ・グリッド・コンピューティングは、識別、認証、認可、機密性、完全性、可用性、否認防止、監査といった領域の従来のセキュリティの原則と制御の必要性を否定するものではない（これらの領域は今ここでいくつかの例を挙げたにすぎない）。従来のエンタープライズ・アーキテクチャで必要だったセキュリティの原則と制御は、エンタープライズ・グリッド環境でもやはり必要である。

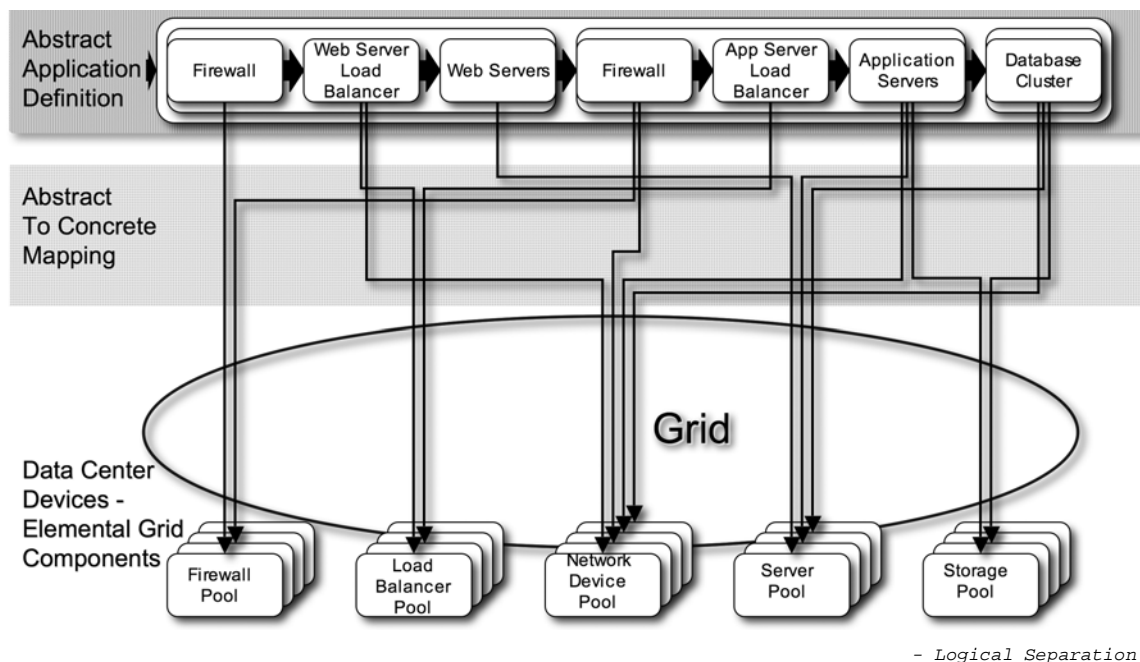


図2 - 抽象コンポーネントから具体コンポーネントへのマッピング2

図2は、物理的に切り分けられた個々のコンポーネントが物理グリッド・コンポーネントの共有プールにどのようにマッピングされるかを示した例である。この場合にセキュリティ上の問題が生じる。この問題は特に新しいものではないが、グリッド環境では大きな問題となる。物理的にサイロ型になっている環境から、コンポーネントがプール化され、多くの場合は相互に関係のない複数のアプリケーションやサービスに共有される環境に移行するときには、プール化されたリソースに対して、異なるアプリケーション毎に論理的に分離された環境とセキュリティを規定する必要がある。たとえば、ストレージ・リソースには、単一のアプリケーションによってのみアクセスされるべき機密情報が含まれる場合があるが、実際の物理ストレージ・プールは他のアプリケーションによっても使用されていることがある。

この種の論理的分離環境はエンタープライズ・グリッド環境に限った固有のものというわけではないが、ここで問題提起する価値はある。同様に、共有リソース上で実行されるアプリケーションやサービスでも、必要に応じてネットワーク・トラフィックを分離する必要がある。たとえば、人事アプリケーションと他のアプリケーションやユーザー・コミュニティが同じ物理システム、ネットワーク、あるいはストレージといったリソースを共有していても、人事部門のネットワーク・トラフィックは人事アプリケーションによってのみ認識され、受信されるべきである。実際、サービスおよび求められる信頼性や保証のレベルに対して定義された要件によっては、特定のコンポーネントが物理的または電氣的に分離される必要があることもある。このように、エンタープライズ・グリッド・アーキテクチャのポリシーの定義と施行メカニズムには、専用リソースを装備できるだけの柔軟性もなければならない。

これは物理リソースだけの問題ではないことに留意されたい。リソースのプール化によって浮上したセキュリティ上の懸念は、共有サービス（ネーミング、ディレクトリ、時刻、ロギングなどのサービス）のような論理リソースや、ソフトウェア・ベースのクラスタ化メカニズム、パケット・フィルタ、侵入検知システムなどのような制御にも同様に当てはまる。たとえば、すべてのディレクトリを共有ディレクトリ・サーバー・プールに入れるよりも、ディレクトリのツリーまたはエントリの一部を他と分離することのほうが有用であったり、そうすることが必要となることもある。

この例は、前述の様々な区分けのタイプ、最小特権、縦深防御を実現しやすくするための、柔軟で拡張可能なセキュリティ・ポリシーの定義と施行フレームワークが必要であることを強調するものとなっている。これもまたエンタープライズ・グリッド・アーキテクチャ固有のものではないが、グリッドの集中管理フレームワークは、一元化されたポリシーのデプロイメントと施行を単純化するのに役立つ。

構成要素から成るサービスのレベルから論理/物理グリッド・コンポーネントの1つ1つのレベルに至るまで、グリッド・コンポーネントには、セキュリティのプロパティまたは属性が関連付けられている。これらのセキュリティ属性は、グリッド・コンポーネントの内部属性（たとえばファイル・システム内のファイル許可）である場合や、グリッド管理エンティティ（次のセクションを参照）によって管理対象グリッド・コンポーネントに明示的に関連付けられた属性である場合がある。後者のタイプの属性は、従来のデータ・センター環境よりもグリッド環境によく見受けられる。

コンポーネントにはさらに、特定の依存関係を定義できる。こうした依存関係は、そのコンポーネント内の特定の属性に対して設定することもでき、また、外部要素（サービスやサービス属性など）に対して設定することもできる。これらの依存関係を使用すれば、セキュリティ・ポリシーの施行を促進でき、機密漏れを最小限にとどめることができる。たとえば、前出の図2の抽象アプリケーション定義部分で表されるサービスに対して、Webサーバーはそれをホストするリソースが十分に安全になるまで（さらに検証されるまで）開始されないという依存関係を宣言することが考えられる。同様に、ネットワークのセキュリティ・ポリシーが適切に施行されるようにするために、Webサーバーがオンラインになる前にファイアウォールのプロビジョニングが行われて用意ができていることを要求する別の依存関係も考えられる。

エンタープライズ・グリッド全体に渡る依存関係と制約の概念は、サービスやビジネス機能全体の安全なプロビジョニング、構成、起動を可能にする。関係するあらゆるステップでセキュリティの属性、依存関係、制約を規定することによって、リスクを最小限にし、機密漏れをある一定の範囲内にとどめることができる。しかし、この方法自体にもリスクがないわけではない。属性の構成が不十分だったり依存関係が合っていなかったりすると、環境にセキュリティ・ホールができたり、サービスの適切な起動や使用ができなかったりする場合がある。したがって、そうした問題を可能な限り検出するために、データの健全性と妥当性の検査機能を実装することが不可欠である。自動化できない点は追加でガイダンス資料を用意し、関係者の意識向上、訓練、実施プロセスにおいて徹底する必要がある。

3.1.1 グリッド・コンポーネントのライフサイクル

EGA 参照モデルでは、グリッド・コンポーネントのライフサイクルの状態は次のように定義されている。

- プロビジョニング
- 継続的管理
- デコミッショニングおよび目的の再割り当て

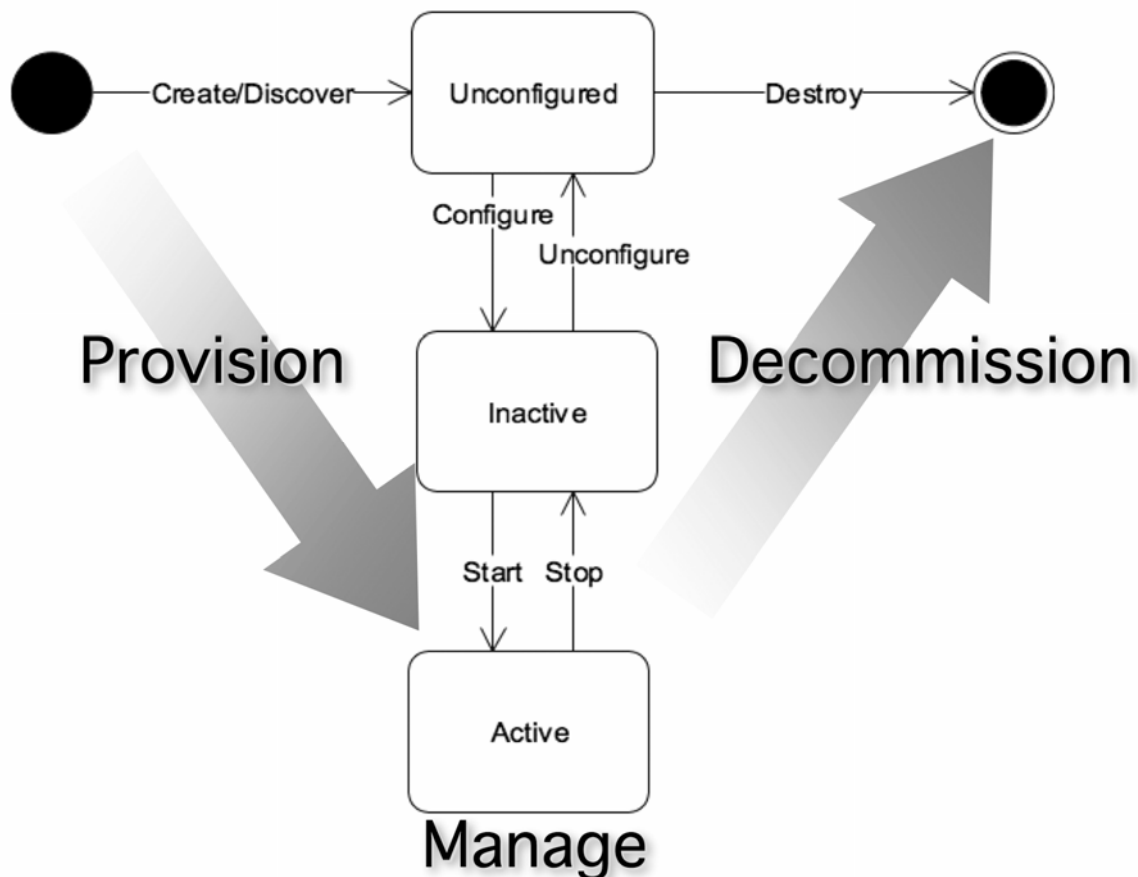


図 3 - 一般的なグリッド・コンポーネントのライフサイクル

図 3 は、これらの状態と状態間の遷移を示したものである。それぞれの状態ごとに、以下のセクションで述べる考慮すべきセキュリティの属性とプロパティがある。

3.1.1.1 プロビジョニング

グリッド・コンポーネントのプロビジョニングには、コンポーネントの追加または作成、構成、アクティブ状態にするための開始が含まれる。検討すべきセキュリティ上の課題として、以下のことがある。

- だれ（ユーザーまたはアプリケーション/サービス）がそのグリッド・コンポーネントのプロビジョニング（作成/検出）を行えるか。
- だれがそのグリッド・コンポーネントのプロビジョニングを試みたか。
- だれがそのグリッド・コンポーネントのプロビジョニングを行ったか。
- いつ、（課金目的で）グリッド・コンポーネントのプロビジョニングが行われたか。
- グリッド・コンポーネントのプロビジョニング履歴（監査記録用）はどうなっているか。
- プロビジョニング前のグリッド・コンポーネントはどんな状態にあったか（新規インストール、目的の再割り当て、不明など）。

- グリッド・コンポーネントはプロビジョニング後に検査され、期待される状態（構成や実行時間）になったことが確認されたか。
- プロビジョニング対象のソフトウェア・イメージは信頼できるものか。その出所は信頼のおけるものか。イメージの完全性は検査されたか。悪意のあるコードは含まれていないか。
- 必要な依存関係がプロビジョニング前にすべて満たされているか。
- 与えられた目的にグリッド・コンポーネントを使用できなくなるような制約はないか（安全ではない場所に配置された、リソースやセキュリティ機能が不足している、現在問題を調査中である、など）。
- プロビジョニングが行われたグリッド・コンポーネントのセキュリティの状態と完全性が、起動前に検証されたか。
- プロビジョニングの優先順位の設定（高優先順位のリクエストは低優先順位のリクエストより先に実行できるような設定）。
- グリッド・コンポーネントがそれまでに使用されていた場合、機密データが残って次のユーザー、アプリケーション、またはサービスに見られることのないように、そのグリッド・コンポーネントのデータ消去が行われたか。消去すべきデータには、たとえば、フラッシュ PROM や BIOS の設定、オペレーティング・システム、アプリケーション、ユーザーの構成およびデータのオブジェクトなど、グリッド・コンポーネントのデコミッションングおよび（再）プロビジョニング前の元の目的に関連した情報が含まれる。

3.1.1.2 継続的管理

グリッド・コンポーネントの継続的管理には、コンポーネントがアクティブ状態にある間の管理に関連したアクティビティが含まれる。プロビジョニングが行われていない（したがって非アクティブ状態にある）コンポーネントに対しては、プロビジョニング以外の管理機能は実行できない。検討すべきセキュリティ上の課題として、以下のことがある。

- だれが管理役割を作成、変更、または削除する権限を持っているか。
- どこから管理者がグリッド管理機能を実行できるか。管理者がエンタープライズ・グリッドに接続する場所や、エンタープライズ・グリッドに対する管理者の認証方法によって、管理者が実行できるアクションの制限はあるか。
- だれが、グリッド・コンポーネントを作成、変更、または削除する権限を持っているか。
- どのような管理役割が作成、変更、または削除されたか。それはいつ、だれによって行われたか。
- だれ（ユーザーまたはアプリケーション/サービス）が、グリッド・コンポーネントとそのセキュリティ属性を管理できるか。
- だれが、グリッド・コンポーネント間の関係を定義できるか。
- だれが、グリッド・コンポーネント間の依存関係やグリッド・コンポーネントの制約を定義できるか。
- グリッド・コンポーネントのどのようなセキュリティ属性を管理できるか。
- セキュリティのどのような属性、関係、依存関係、制約が変更されたか。それはいつ、だれによって行われたか。

- ヘテロジニアスな環境で、どのようにグリッド・コンポーネントのセキュリティの属性とポリシーを安全に配布または更新できるか。また、ヘテロジニアスな環境で、どのようにセキュリティ・ポリシーがグリッド・コンポーネントによって一貫性をもった形で解釈されるか。
- 様々な製品のセキュリティ属性を記述するための共通タクソノミはあるか。製品タイプ毎（たとえば Web サーバーやディレクトリ・サーバーなど）に定義できる共通属性はあるか。セキュリティ・ポリシーの枠組みの中で製品固有の属性がどのように追加され管理されるか。
- グリッド管理エンティティは、要件を定義したり意思決定したりするときに、共通属性とベンダー/製品特有の属性の両方を使用できるか。
- デプロイメントされた各グリッド・コンポーネントのセキュリティ構成はどのように検証されるか。こうした検証に使うべき標準的な形式はあるか。広範な製品に及ぶ整合性に関する検証は、個々の製品として、およびそれによって構成されるリソースの一部としてどのように評価されるか。
- 障害時に、それに関連したどのような悪影響が生じるか。セキュリティ上の障害の検出と回復を行うための自動化された、または手作業によるプロセスはあるか。自動化が可能な場合、どのようなレベルの特異性まで示されるか。セキュリティ上の障害（およびその後の回復アクション）が管理者にどのように通知されるか。
- グリッド・コンポーネントの変更は、グリッド管理エンティティを介してのみ行われるべきであるが、それ以外による変更がどのように検出され訂正されるか。
- グリッド管理フレームワークそのものに対する変更は、どのように防止され検出されるか。グリッド・フレームワークがセキュリティ侵害やシステム保全性の違反を検出した際、どのような対応を行うべきか。
- グリッド管理フレームワークが、エンタープライズ・グリッド・アーキテクチャ上で発生するセキュリティ・イベントを正規化し、相互に関連付け、報告できるようにするには、どのような機能が必要か。
- 組織のリスク管理の決定に応じて、グリッド管理フレームワークそのものを複数の層や区分に分割することもできるか。
- だれが、セキュリティ・イベントを精査して対処する責任を負っているか。
- どのようにユーザーがグリッド・コンポーネントに認証されるか。
- どのようなメカニズムを使用してユーザーはグリッド・コンポーネントにアクセスするか。
- グリッド・コンポーネントへの出入りを行うに際して、どのようなタイプのネットワーク通信が許可されるか。
- だれがまたは何が、グリッドを辿るときに許可されるネットワーク通信のタイプを決定するか。
- グリッド・コンポーネントのセキュリティの抜け穴がどのような形で含まれるか。

3.1.1.3 デコミッションングおよび目的の再割り当て

グリッド・コンポーネントのデコミッションングは、様々な理由で行われる。サービスを中止したり、使わなくなったグリッド・コンポーネントを削除したりするために、グリッド・コンポーネントの完全なデコミッションングを行う場合がある。また、グリッド・コンポーネントの目的を再割り当てし、さらにもう一度プロビジョニングを行えるようにデコミッションングを行う場合もある。検討すべきセキュリティ上の課題として、以下のことがある。

- だれ（ユーザーまたはアプリケーション/サービス）が、そのグリッド・コンポーネントのセキュリティ属性のデコミッシング/目的の再割り当て（作成/検出）を行えるか。
- だれがリソースのデコミッシング/目的の再割り当てを試みたか。
- どのようなリソースのデコミッシング/目的の再割り当てが行われたか。それはいつ、だれによって行われたか。
- いつ、（課金目的で）リソースのデコミッシング/目的の再割り当てが行われたか。
- リソースのプロビジョニング/目的の再割り当て/デコミッシングに関する履歴（監査記録用）はどうなっているか。
- リソースのデコミッシング/目的の再割り当てが行われる前の必要な状態が記録されているか（たとえば監査ログやシステム・ログ、暗号鍵データ、訴訟用データなど）。
- どのような条件でリソースのデコミッシング/目的の再割り当てが行えるか。グローバルな制約、または場所、サービス、ユーザーなどに特有な制約はあるか。
- 必要な依存関係がデコミッシング/目的の再割り当ての前にすべて満たされているか。
- デコミッシング/目的の再割り当ての前にリソースの浄化（Sanitize）が行われているか。使用される浄化手順に関して、必要な保証レベルが満たされているか。機密データが残って次のユーザー、アプリケーション、サービスによって見られることのないように、グリッド・コンポーネントのデータ消去を行う必要がある。消去すべきデータには、たとえば、フラッシュ PROM や BIOS の設定、オペレーティング・システム、アプリケーション、ユーザーの構成およびデータのオブジェクトなど、グリッド・コンポーネントのデコミッシング前の元の目的に関連した情報が含まれる。データ消去のレベルは、そのグリッド・コンポーネントの目的の再割り当てが行われるのか、それがグリッドから完全に削除されるのかによって異なる場合がある。

3.2 グリッド管理エンティティのセキュリティ

EGA 参照モデルでは、**グリッド管理エンティティ (GME)** は、以下のものを管理する**論理エンティティ**と定義されている。

- グリッド・コンポーネント
- グリッド・コンポーネント間の関係
- グリッド・コンポーネントのライフサイクル全体（プロビジョニングからデコミッシングまで）

GME は、人的リソース、プロセス、技術の任意の組み合わせとして実現できる。GME は、論理的には管理対象となるグリッド・コンポーネントとは区別されているが、GME 機能の実現は、グリッド・コンポーネント自体の実現とはそれほど明確には分離できない場合もある。図 4 では、GME を、管理されるグリッド・コンポーネントとは論理的に別個のエンティティとして示しているが、実際は、グリッド・コンポーネントに管理（GME）機能が含まれる場合もある。したがって、GME は基本的な性質として分散型の階層構造を持っており、アプリケーションやサービスと同様、GME も分解できる（つまりグリッド・コンポーネントに分解できる）。ただし、グリッド・コンポーネント自体が GME 機能を実行する場合は、ある程度注意を払う必要がある。つまり、それによって管理機能の区分化ができなくなり、コンポーネントのセキュリティ障害がその管理機能の妥当性に悪影響を及ぼしたり、さらに GME 全体に影響したりすることもあるからである（たとえば、セキュリティ障害が他の GME 機能への無認可アクセスになる場合など）。

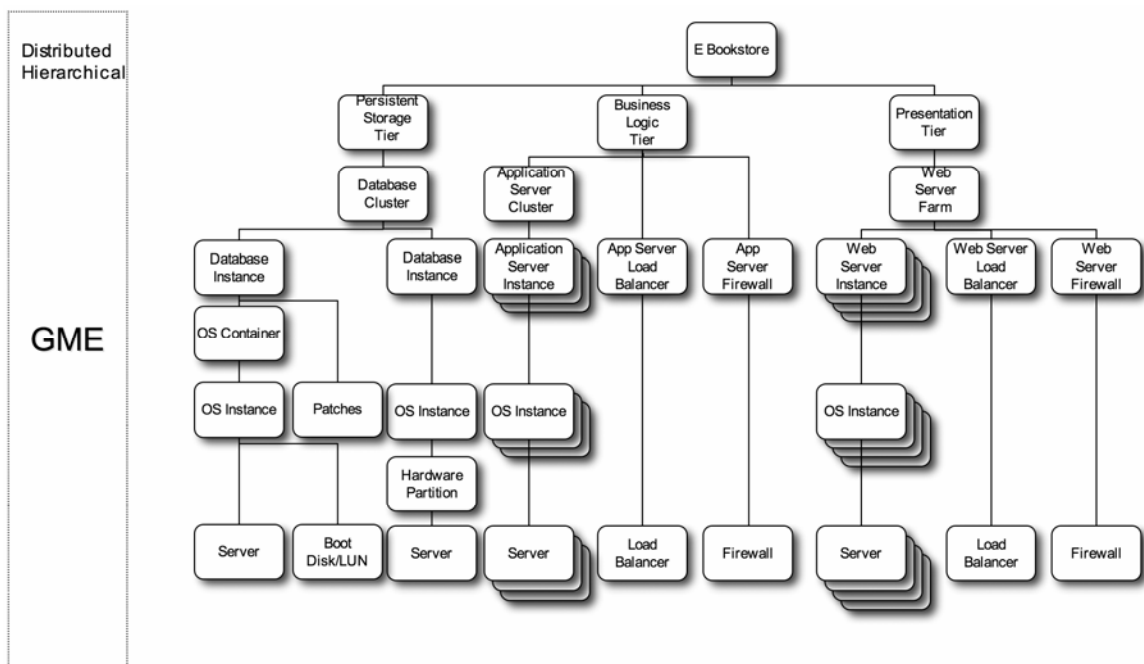


図 4 - グリッド・コンポーネントと論理的に区別される GME

GME での管理における重要な機能の 1 つは、エンタープライズ・グリッドのセキュリティ・ポリシーの定義、施行、検証のサポートにある。グリッド・コンポーネント、コンポーネント間の関係、コンポーネントのライフサイクルの管理に加えて、GME は以下のセキュリティ機能を管理する。

- すべてのユーザーID と管理者の役割の管理
- エンタープライズ・グリッド全体にわたって共有されるすべてのコア・サービスの管理
- (ユーザー、アプリケーション、サービスに関する) アイデンティティの認証
- 認証された対象、非認証の対象によって起こされるアクションの認可
- GME および他のすべてのエンタープライズ・グリッド・コンポーネントへのアクセスの制限
- セキュリティと監査に関係のあるすべてのイベントの収集、保管、分析、報告

- エンタープライズ・グリッド・アーキテクチャ内の要素によって使用されるすべての暗号鍵や共有秘密鍵などの管理（作成、インストール、ローテーション、検証、保管、破棄など）
- グリッド環境での安全な通信の施行（ローカルまたはリモートの管理者のアクセスを含む）
- 共有されるグリッド・コンポーネントの安全な分離の施行、および物理グリッド・コンポーネントで構成されるサービスの安全な分離の施行
- 組織のセキュリティ・ポリシーに従って、グリッド・アーキテクチャ全体のローカルおよびリモート管理とトラブルシューティング操作が安全なものになっていることを確保すること
- グリッド・コンポーネントの1つ1つまたはグループのセキュリティが、それらのオブジェクトの完全性、属性、依存関係、制約などの観点で期待される状態にあるかどうかを判別するための検証

ここまで個々のコンポーネントに関連したエンタープライズ・グリッド特有の要件を述べたが、これらの要件の大部分はグリッド管理エンティティに集中していることに留意されたい。これは、1つには、ポリシーの定義、施行、検証に関して GME の果たす役割の結果だと言える。グリッドのリソース群（あるいはネットワーク化されたリソースの単なるプール）だけでは、それはグリッド環境に固有なものとはならない。固有なのは、集合体として管理される仕方にある。グリッドに関するリソースのプールのプロビジョニング、管理、デコミッションングの機能を持つ GME の概念を導入することで、グリッド環境に固有な脅威とセキュリティ要件の核心をつかむことができる。これらの課題を理解して対処することにより、GME によるセキュリティ管理の（論理的な）集中化のメリットを活かすことができる。

4 グリッド・セキュリティのユースケース

以下に示すユースケースは、EGA リファレンス・モデル・ワーキンググループによって策定されたユースケースに基づき、現実世界の状況で予想されるセキュリティのリスク、課題、アクションに特に焦点を当てたものとなっている。

以下のユースケースでは、要件やガイダンスを詳述する必要のある、エンタープライズ・グリッド・コンピューティング環境に関連した特定のセキュリティ・リスクを取り上げる。これらのユースケースは、構成要素としてのコンピュータ、ネットワーク、ストレージといったインフラストラクチャが、グリッドがデプロイメントされる環境に対してその安全性が適切に確保されていることを前提としている。こうしたインフラストラクチャ要素を安全にするための推奨事項は、ベンダー、業界、政府機関などの様々な情報源から入手できる。

4.1 汎用ユースケース – グリッド・コンポーネントのプロビジョニング

グリッド・コンポーネントがグリッドに加入する最初のフェーズは、プロビジョニングである。プロビジョニング・プロセスはさらに以下に示す3つのフェーズに分けられ、それぞれをサブ・ユースケースと考えることができる。

4.1.1 グリッド・コンポーネントの追加/作成

このフェーズは、新しいグリッド・コンポーネントの作成を表す。このアクティビティには、以下のアクティビティが含まれる。

- グリッド・コンポーネントがグリッドに結合する。つまり、グリッド・コンポーネントが自動検出されるか手作業で追加されることにより、GME がそのグリッド・コンポーネントを識別して管理できるようになる。グリッド・コンポーネントは適正なコンポーネントでなければならない。適正さを装った悪質なコンポーネントであってはならない。同様に、グリッド・コンポーネントになるものは、適正な GME と通信していることを検証できなければならない。

したがって、この結合または検出プロセスには、GME とグリッド・コンポーネントの間で信頼関係が確立されるある種の相互認証が含まれている必要がある。相互認証の要件は、グリッド・コンポーネントを装ったものを阻止するためのリスク軽減制御を適所に配置した環境を実現できるような仕方で構成できる必要がある。

- グリッド・コンポーネントの新しいインスタンスは、論理的に管理可能な既存のグリッド・コンポーネントから、（依存関係を使用して）自動的に生成されるか、手作業で作成される。既存のコンポーネントは、GME によってすでに認証されているか、あるいは他の形で信頼されていなければならない。新しいインスタンスも、それ自体グリッド・コンポーネントなので、認証される必要がある。

グリッド・コンポーネントが検出された場合であっても、新しく作成された場合であっても、このサブ・ユースケースの結果として、GME はグリッド・コンポーネントを追跡し、その機能だけでなくセキュリティのポリシーやプロパティも確認する。グリッド・コンポーネントのセキュリティのプロパティ、依存関係、制約が不明で検出できない場合、GME はそのセキュリティの要件と制約が定義されるまで、それをアクティブ状態にして使用することを避ける。

4.1.2 グリッド・コンポーネントの構成

このフェーズは、与えられたコンポーネントをアクティブな状態にできるように構成するアクティビティを表す。グリッド・コンポーネントの構成に使用される属性や他の要素は、手作業で割り当てられるか、GME が提供するインタフェースを使用して検出される。セキュリティを構成するある特定の要素がコンポーネントに適用できない場合や、指示されたセキュリティ・ポリシーをコンポーネントに実装できない場合、GME はそのコンポーネントをアクティブ状態にして使用することを許可しない。コンポーネントの使用を許可したりそのセキュリティ・ポリシーや構成を調整したりするには、管理上の介入が必要となる。構成アクティビティには、以下のことが含まれる。

- コンポーネントの基本的なセキュリティ要件の定義または検出
- セキュリティの依存関係と制約の定義または検出
- コンポーネントのユーザーと管理者に関するアクセス制御ポリシーの適用
- システム、ネットワーク、ストレージの通信アクセスパスの定義
- 他のグリッド・コンポーネントとのセキュリティ・パラメータの結合
- コンポーネントのセキュリティ・ポリシーが正しく適用されており、必要な依存関係がすべて満たされていることの検証。
- SLO を定義するセキュリティ設定の指定。これには、セキュリティの「強化」や「封鎖」要件が含まれる（Web サーバー・サービスで使用可能なポート/プロトコル、使用可能 o/s サービスなど）。

4.1.3 グリッド・コンポーネントの開始

このフェーズは、コンポーネントを有効にし、エンタープライズ・グリッドに配置してアクティブ状態にすることにより、それを使用できる状態にするアクティビティを表す。このアクティビティは、適切に認証されて認可されたユーザーまたはサービスのみが GME を介して実行できる。さらに、この操作を適用できるのは、プロビジョニングが行われ、（上記の構成ステップに従って）セキュリティ・ポリシーが検証されたグリッド・コンポーネントに対してのみである。同様に、グリッド・コンポーネントを有効にして使用できるようにする前に、必要な依存関係と制約がすべて満たされていないなければならない。

ここで留意すべきなのは、ユーザー、管理者、サービスが、直接グリッド・コンポーネントを開始または停止することを許可されない場合がある。組織のポリシーによっては、GME が、これらの機能を実行するときにアクセスされなければならないアーキテクチャ上のハイパーバイザー（Hypervisor）として使用される場合がある。この場合は、管理者が、様々なコンポーネント操作を実行するために、GME との間に入ってそれらの操作を指示するインタフェースとしての役割を果たす。GME は、管理者に代わって実際のコンポーネント操作を実行することになる（このとき、管理者はそのアクションを実行できる適切な権限を持っているものとする）。このモデルには、ユーザーやサービスが他のグリッド・コンポーネントを管理するために直接アクセスすることを許可しない、という利点がある。さらに、このような機能で GME を使用すると、すべての管理操作が検証され監査されるので、より大きな運用上の保証が得られるようになる。最後に、このモデルには、GME が障害を起こしたコンポーネントを自動的に再始動できるという別の利点もある（これが行えるかどうかはポリシーに依存する）。

4.2 汎用ユースケース – グリッド・コンポーネントのモニターと管理

モニターと管理のユースケースには、各グリッド・コンポーネントに対する継続的なセキュリティ管理が含まれる。これは、グリッド・コンポーネントのプロビジョニングが正常に行われ、GME、グリッド・コンポーネント、そしてその構成要素の間に信頼関係が確立されていることが前提となる。この信頼関係があることで、GME はモニターが行える、つまりグリッド・コンポーネントに関する情報へのアクセスが認可される。この情報は機密情報の場合があり、おそらく公開情報ではないが、他のグリッド・コンポーネントと共有される（必ずしも知る必要のない）ような情報の場合もある。このユースケースには、GME からグリッド・コンポーネントに対する管理コマンドも含まれる。これらのコマンドは、グリッド・コンポーネントがコマンドの出所を信頼するように安全なものになっていなければならない。セキュリティに関係のあるモニターと管理のアクティビティとして、以下のことが考えられる。

- GME からモニターや管理が行える属性の自動検出と手作業による定義。
- セキュリティに関係のある計測指標とイベントのモニター（たとえば、グリッド・コンポーネントの利用と可用性、アプリケーションへのユーザー/サービスのログオンとログオフの試み、グリッド上で実行されているサービスまたは GME など）。侵入の検出と防止の機能は、このカテゴリに分類される。
- グリッド・コンポーネントの監査およびセキュリティのすべてのイベントのロギング、保管、分析、関連付け。GME は、チャージ・バック（chargeback）、整合性（compliance）、監査の要件などのために、使用量を測定して記録することが必要な場合もある。監査ログは、無許可の開示、変更、破棄から保護されなければならない。
- セキュリティ・パッチの状況の報告と、セキュリティ・パッチの適用（GME を介して）。
- 無許可で、あるいはグリッド・ポリシーに違反してエンタープライズ・グリッドにインストールされたコンポーネント、そこで変更されたコンポーネント、そこから削除されたコンポーネントの識別。
- GME によって実行されるその他のシステム管理アクティビティやアイデンティティ管理アクティビティ。

4.3 汎用ユースケース – グリッド・コンポーネントのデコミッションング

デコミッションング・プロセスは以下に示す 3 つのフェーズに分けられ、それぞれをサブ・ユースケースと考えることができる。

4.3.1 グリッド・コンポーネントの停止

このアクションは、認可されたユーザーまたはサービスに代わって GME によって実行される。このアクションは指定されたグリッド・コンポーネント（およびその構成要素）を無効にするだけでなく、そのコンポーネントに依存するコンポーネントも無効にする場合がある（それが適当な場合）。無効にされたグリッド・コンポーネントは、使用できなくなる。そのコンポーネント上の既存のユーザー操作やトランザクション操作は、可能な場合は他のリソースに転送される（これはサイト・ポリシーに従う）。そのようにしてコンポーネントが静止状態になったところで、そのコンポーネントは停止する。これとは別に、GME が即時停止の概念をサポートする必要がある場合もある。即時停止では、グリッド・コンポーネントは既存のサービス状態、ユーザー・データ、トランザクションの保存を試みないで停止する。グリッド・コンポーネントは、停止後も GME が管理可能な状態にある。

4.3.2 グリッド・コンポーネントの構成解除

グリッド・コンポーネントの完全なデコミッションングを行う前に、まず構成解除を行う必要がある。このステップに関連するアクティビティとして、重要な状態の保存とコンポーネントの浄化がある。セキュリティ・ログやトランザクション・ログ、暗号鍵データ、さらにはイベントのキー・コンポーネントの構成とデータ・ファイルといった重要な状態は、将来の使用のために、あるいは監査または法的調査をサポートするために保存する必要がある場合がある。したがって、コンポーネントを構成解除する前に、まずその重要な状態を収集しなければならない。

重要な状態に関するデータの保存が完了したら、次はコンポーネントの浄化を行う。セキュリティの観点から、グリッド・コンポーネントのすべての機密情報のデータ消去を行い、関連するセキュリティのポリシーとプロパティを GME から削除する必要がある。また、そのコンポーネントを将来使用できないように防御またはロックしているセキュリティのポリシーとプロパティはリセットする必要がある。

構成解除されたグリッド・コンポーネントは、（次の項で説明するように）グリッドから削除するか、目的の再割り当てと再プロビジョニングを行って他の機能に使用できる。後者の場合のグリッド・コンポーネントの構成解除の程度は、その目的がどのように再割り当てされるかによって決まる。たとえば、Web サーバーを実行するコンピュータ・コンポーネントを、別のアプリケーションまたはサービスのための Web サーバーとして実行するように再プロビジョニングを行うのは簡単である。一方、この同じコンピュータ・コンポーネントを、アプリケーション、それが動作するネットワーク、ユーザーのいずれも異なるデータベース・サーバー用に再プロビジョニングすることもできるが、このためには大がかりな浄化が必要となる。

4.3.3 グリッド・コンポーネントの削除

浄化が行われたグリッド・コンポーネントは、グリッドから削除してデコミッションング・プロセスを完了する必要がある。こうすることで、そのコンポーネントが何らかの使用目的で別のグリッド・コンポーネントによって再割り当てされるのを防止する。削除されたグリッド・コンポーネントは、グリッドから完全に分離される。この状態のコンポーネントは、エンタープライズ・グリッドのアクティビティに参加することは許可されない。このコンポーネントをグリッドに再結合するためには、再度それを追加し、グリッド・コンポーネントの通常の作成/追加プロセスを実施する必要がある。削除されたグリッド・コンポーネントについては、GME による制御もモニターも行われないうちに留意されたい。結果として、このコンポーネントは非グリッド環境のアプリケーションに使用される場合がある。このコンポーネントをグリッドに追加し直す場合は、このことを考慮に入れる必要がある。

4.4 運用ユースケース

このユースケースは、セキュリティに関係のある継続的なイベントに焦点を当てたものである。これらのイベントは、グリッド・コンポーネントの3つのライフサイクルについての前述のユースケースと直交する関係にある。このユースケースは、以下のサブ・ユースケースに分けられる。

4.4.1 ユーザー/サービスのセキュリティ・イベント

このフェーズには、ユーザーおよびサービスの、以下のような基本セキュリティ・アクションが含まれる。

- ユーザー/サービスのグリッド・ログオン – ユーザーまたはサービスが、GME に対して認証を試みる（結果は、成功または失敗となる）。
- ユーザー/サービスのグリッド・ログオフ – 認証されたユーザーまたはサービスが、グリッドからのログオフを試みる。
- ユーザー/サービスの役割代行 – 認証されたユーザーまたはサービスが、別のアイデンティティまたは役割を引き受けることを試みる。したがって、この場合、この別のレベルの特権を得るために GME による再認証が必要である。その際の認証メソッドは、ユーザーまたはサービスが GME に対する最初の認証に使用していた元々の認証メソッドとは異なるものである必要がある場合がある。
- ユーザー/サービスのログオン状況 – （適切な権限を持つ）認証されたユーザー、役割、またはサービスが、特定のユーザーまたはサービスが GME に正常に認証されたかどうかを判別することを試みる。

4.4.2 ワークロードのセキュリティ・イベント

このサブ・ユースケースは、ワークロードつまり「作業単位」の観点でセキュリティを捉えたものである。ここでは、グリッド・コンポーネントは単一のアプリケーションまたはサービスのために専用のものではなく、また、それによってのみ所有されないことを前提としている。このフェーズには、ワークロードに関する以下のセキュリティ・アクションが含まれる。

- ワークロードのサブミット – 認証されたユーザーまたはサービスが、ワークロードのサブミットを行い、実行依頼を試みる。このアクションでは、ユーザーがワークロードを実行依頼してよいかどうか、また、ワークロードがどのようなクラスに入るか（そのような分類が定義されている場合）について考慮すべきである。さらにこのアクションでは、与えられたワークロードをどこで実行できるか、中断して再開することが可能かどうか、移動できるかどうか、いつ開始または停止しなければならないか、許される実行時間、などの関連した属性を限定する場合がある。
- ワークロードの終了 – 認証されたユーザーまたはサービスが、既存のワークロードの終了を試みる。このアクションでは、ユーザーがワークロードを終了できるかどうかやワークロードのクラスについて考慮すべきである。また、このアクションでは、特定のタイプのワークロードを終了できるかどうか、あるいは、いつそれを行えるかについての制限を設ける場合もある（整合性と完全性を確保する目的で）。
- ワークロードのモニタリング – 認証されたユーザーまたはサービスが、既存のワークロードの進行状況/状態のモニタリングを試みる。このアクションでは、ユーザーまたはサービスが既存のワークロードをモニターするのに必要な権限を持っているかどうかについて考慮すべきである。また、このアクションでは、リクエストのクレデンシャルに基づいてどのような情報を問い合わせることができるかの制限を設ける場合もある。
- ワークロードのスケジューリング – 認証されたユーザーまたはサービスが、ワークロードまたは一連のワークロードの実行のスケジューリングを試みる。「ワークロードのサブミット」のアクションの場合と同様、このケースでも、どのようなワークロードを実行できるか、それらをどこでどのような順序で実行できるかを判別する。このアクションは、スケジューリング優先順位やリソースの可用性に基づいたワークロードの中断と再開を担う場合がある。さらに、このアクションでは、（同一リソース上またはグリッド全体では）同時に実行できないワークロード（またはワークロードのクラス）を明示的に宣言する場合がある。運用上の競合を防止したり、ユーザー、サービス、あるいは「顧客」の分離を促進したりするためには、これが必要である。
- ワークロードのバッチ処理 – GME の責務。このアクションは、グリッドにおけるワークロードの振る舞いの制御を担う。特に、このアクションは、リソースへのワークロードの転送、リソースの結果の収集（そのような結果がある場合）、および開始、停止、中断、再開、状態問い合わせなどのすべてのワークロード制御/モニター機能を担う。

5 グリッドの脅威とリスク

エンタープライズ・グリッド・コンピューティングは、エンタープライズ IT 環境の視覚化、集約、仮想化、プロビジョニング、管理のための新しい方法を提供するものである。この新しいモデルに基づくことによって、企業は、今日のデータ・センターと企業内ネットワークの枠組みでは通常は実現できない仕方で、IT 資産の標準化、自動化、最適化を行うことができる。こうしたメリットはあるものの、このアーキテクチャのデプロイメントを行うことのもリスクも考慮しなければならない。この新たなリスクに対処するためには、まず、エンタープライズ・グリッド・コンピューティング・アーキテクチャに特有な脅威をよく理解する必要がある。

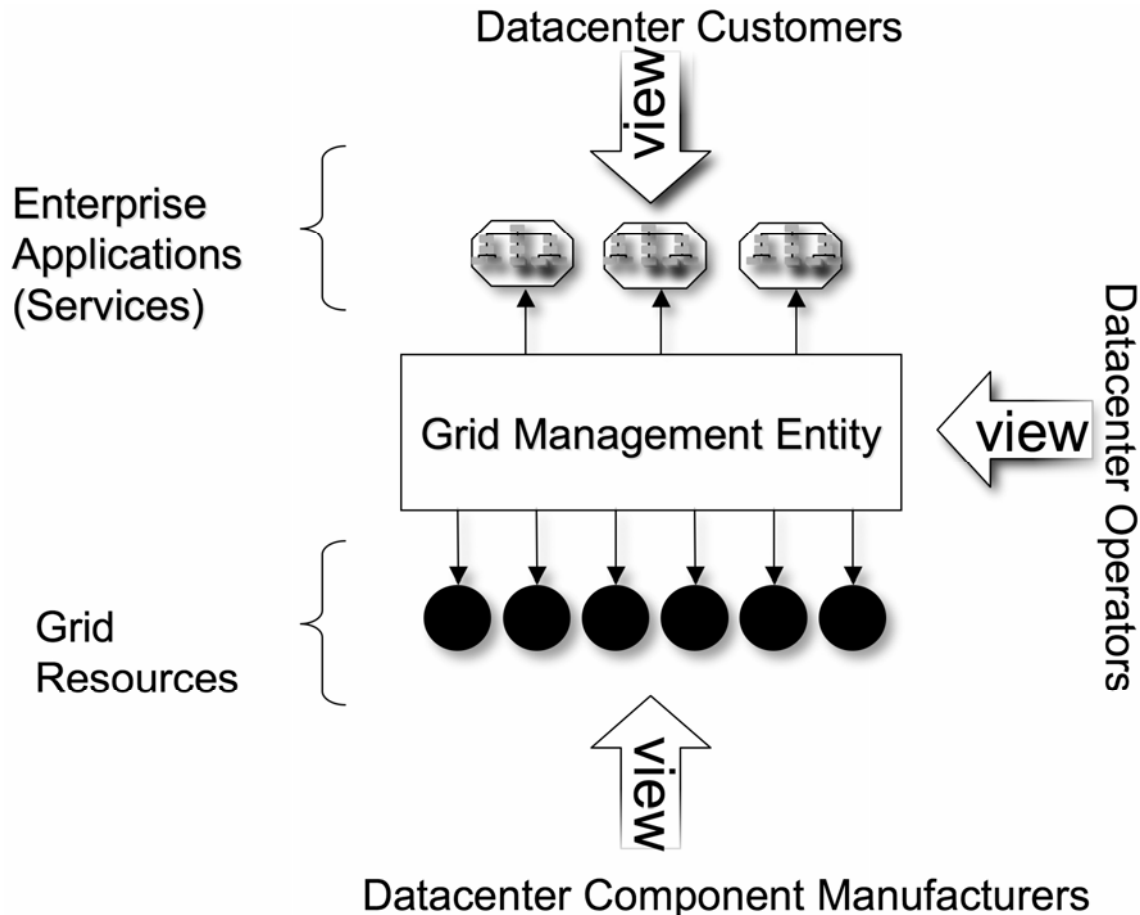


図 5 - データ・センターの基本的概念

エンタープライズ・グリッド・コンピューティングがもたらす新しいセキュリティの懸念は、GME が個別のグリッド・リソースやグリッド・リソースのグループとどのように通信して、図 5 に示すように 1 つのまとまりとして協働させるのかというところに端を発している。GME は、すべてのグリッド・リソースを認識し、それぞれのグリッド・リソースに対していつ何をすべきかを伝えることができるという点で強力なエンティティである。また、GME は、グリッドに複数のワークロードが与えられたときの公平性も実現する。つまり、どのユーザーのワークロードでもグリッド上で実行されるチャンスが公平に回って来るよう調整するのは GME である。さらに、セキュリティ・ポリシーの観点からグリッド・コンポーネントの整合性の強化、監査、検証を行うのも GME である。

エンタープライズ・グリッドには、定期的に目的の再割り当てや再プロビジョニングが行えるグリッド・リソースの柔軟な共有プールを使用するという特徴もある。したがって、セキュリティの脅威とリスクを考えると、エンタープライズ・グリッド・コンポーネントのライフサイクルも考慮に入れる必要がある。

以下に示す脅威とリスクのカテゴリは、エンタープライズ・グリッド固有の特性に基づいたものである。このリストにすべてが含まれているわけではないが、ここでは、サンプルとして、この種のアーキテクチャに本質的に潜むようなリスクと脅威のタイプの中で代表的なものを記載している。

- アクセス制御に対する攻撃。これは、無認可エンティティや認可エンティティが、アクセス制御ポリシーを無視したり破ったりすることに関連するリスクを表す。
 - 無認可のグリッド・コンポーネントまたはユーザーがグリッドに参加する。
 - 無認可のユーザーまたはサービスが、グリッドのアプリケーションおよびサービスの実行依頼、終了、制御、モニタリングを試みる。
 - 認可されたユーザーまたはサービスが、グリッド上で実行されるアプリケーションやサービスのアクセス制御を無視したり、しようと試みる。
 - グリッド上で実行されるユーザー、アプリケーション、またはサービスが、許可された特権またはリソースの範囲を超えようとする。
- グリッドの監査システムおよび課金システムの破壊。これは、エンタープライズ・グリッド環境固有の監査システムおよび課金システムの完全性に対する脅威を表す。これには、偽のイベントを投入することや、オーバーフロー、イベント変更、その他監査システムに対してよくある様々な攻撃が含まれる。
- サービス妨害 (DoS)。これは、サービスまたはリソースの可用性に関するあらゆる種類の攻撃のことである。エンタープライズ・グリッドでは一般に、非グリッド環境に比べて高い可用性が得られるが、リスク・アセスメントの中で次の DoS 脅威を考慮に入れる必要がある。
 - GME に対する DoS 攻撃。
 - 新しく認可されたグリッド・コンポーネントまたはユーザーがグリッドに正常に結合されないよう、グリッド・コンポーネント結合プロトコルに対して行なわれる DoS。
 - 認可されたグリッド・コンポーネントまたはユーザーが、グリッドからの退去を「余儀なく」される。
 - ユーザーまたはサービスがグリッドをワークロードで超過させようとする。これによって、コンピュータ、ネットワークおよび/またはストレージ・コンポーネントが使い尽くされ、リソースへのアクセス待ち時間が他のグリッド・ユーザーに著しい影響を及ぼす。
 - GME からグリッド・コンポーネントへのスケジューリング・メッセージを無効化または変更して、あるアプリケーション/サービスの優先順位を他のものより不当に高くする。
 - グリッド全体およびグリッド全体の QoS に影響を及ぼすその他の DoS 攻撃。
- 悪意のあるコード、マルウェア (malware)。これは、グリッド環境に無認可で侵入する、自分の特権を引き上げる、自分の存在を隠蔽する、有効なコンポーネントを偽装する、増殖するなど、明らかにエンタープライズ・グリッド・アーキテクチャのセキュリティ・ポリシーに違反していることを試みるコードのことである。

- オブジェクトの再利用。これは、無認可ユーザーが機密データを入手できるようにする方法である。エンタープライズ・グリッド環境では、グリッド・コンポーネントのデコミッションング（および浄化）が適切に行われないと、これはリスクとなる。
- 偽装攻撃。これは、有効なグリッド・コンポーネントが騙されて、有効なグリッド・コンポーネントを装った別のエンティティと通信したり連動したりすることになる類の攻撃である。このような間違いによって、情報の開示や変更、無認可トランザクションの実行などが許可されてしまうことがある。このような侵害の及ぼす影響は、ターゲット・コンポーネントと攻撃側コンポーネントの間の信頼関係と、そのとき施行されているセキュリティ・ポリシーに依存する。
- スニファ（Sniffers）。ネットワーク上を移動するパケットを監視することを、スニフing（sniffing）またはスヌーピング（snooping）という。エンタープライズ・グリッドでは、アプリケーション/サービス、GME、グリッド・コンポーネントの間に追加のネットワーク・トラフィックが発生する可能性があり、これを保護する必要がある。この脅威への対処に失敗すると、データ操作攻撃や反射攻撃など、他のタイプの攻撃を招くことがある。

これまでに挙げた脅威とリスクに加え、エンタープライズ・グリッド環境に固有のものではないが、ここで述べるのがよいと思われる一般的な別のカテゴリがある。

- 物理的セキュリティは、情報システム全般のセキュリティの枠組みの重要な構成要素である。どの情報システムでもそうであるように、エンタープライズ・グリッドは、人為的災害や自然災害だけでなく、人間による（悪意のある、あるいは偶発的な）物理的脅威からも保護する必要がある。複数のサイロ型システムではなく1つのエンタープライズ・グリッド・システムを持つことによって、セキュリティの物理的脅威への対処の効率化につながる。
- ソーシャル・エンジニアリングとは、無認可ユーザーが認可ユーザーを騙してシステムにアクセスするのに必要な情報を入手する方法のことである。エンタープライズ・グリッドは、必ずしもソーシャル・エンジニアリングの新しい形の脅威をもたらすわけではなく、強力な一連のセキュリティ管理（たとえば、明確な管理モデル、プロセス、ポリシー）と一体でエンタープライズ・グリッドの提供を開始すれば、むしろこうした脅威を低減できる。
- 法規制順守を徹底することにより、グリッドとそのユーザーによって実行される機能が行政や業界の規制に違反することがなくなる。規制に含まれるセキュリティ要件は、グリッドのポリシー、手順、プロセスに盛り込む必要がある。

どのような環境にある GME でも、人的リソース、プロセス、技術の組み合わせから、より自動化されたものへと発展するにつれて、これらのリスクの多くがいつそう明確になる。GME におけるエンタープライズ・グリッド管理の集中化（それは論理的なものにすぎないが）は、悪意のあるユーザーにとって魅力的なターゲットになる可能性があるが、逆に、GME を介して集中的な仕方ですべてを安全にするためにより多くのリソースを追加することが可能となる。GME が自動化されるほど、これらのリスクの多くに容易に対処できるようになる。

これまでに挙げたリスクはすべて、全体的なリスク・アセスメントの一部として盛り込む必要がある。そうすることによって、これらのリスクを軽減するか、リスクを分散させるか、あるいはリスクを甘受するかのビジネス上の意思決定を行えるようになる。次のセクションでは、これらのリスクの軽減に使用できるグリッド・セキュリティ要件を詳しく説明する。

6 グリッド・セキュリティ要件

組織のセキュリティ要件は、その組織の固有の側面を計算に入れることができ、またそうすべきである。これらの要件は、組織のセキュリティ・ポリシーに基づく必要がある。セキュリティ・ポリシーは、何らかのリスク・アセスメントの実施から導き出される。エンタープライズ・グリッド環境については、前述の脅威をリスク・アセスメントのベースにすることができる。次に組織は、コスト/メリット/影響を分析した上で、リスクを軽減するか、リスクを分散させるか、あるいはリスクを甘受するかを決定する必要がある。以下に示すセキュリティ要件は、エンタープライズ・グリッド環境に固有の脅威とリスクを軽減するアプローチに基づくものである。

6.1 機密性 (Confidentiality)、完全性 (Integrity)、可用性 (Availability) - (CIA)

どの情報セキュリティ・システムにも当てはまるこの3つの基本要件が、エンタープライズ・グリッド・コンピューティング環境にも適用される。これらの要件の大部分は、各グリッド・コンポーネントの既存のセキュリティ機能によって満たされると考えられる（ただし、これらは必ずしもグリッド環境に固有のものではない）。しかし、これらの要件の中に、グリッド環境に固有の側面が含まれる（これは特に GME の存在による）。

- GME とグリッド・コンポーネントの間の通信、あるいはグリッド・コンポーネントの集合の中での通信は、安全でなければならない。これには、ネットワーク上の改ざんから保護するチャンネル暗号化や完全性検査のような手段で機密性を持たせることが含まれる。また、否認防止要件を満たすこともこれに含まれる（必要な場合）。
- 機密データの機密性は、グリッド・コンポーネントのライフサイクル（デコミッションングから目的の再割り当て、および再プロビジョニングまで）を通して維持されなければならない。
- グリッド・コンポーネントのプロビジョニングに使用されるイメージと構成プロセスの間に使用される設定（属性、依存関係、制約）は、完全性が検証されなければならない。これらの設定の変更は、GME によって管理されなければならない。同様に、違反は GME によって検出されなければならない。完全性を検証できない項目はエンタープライズ・グリッドから分離して、その使用や伝搬を未然に防がなければならない。
- グリッド・コンポーネント自体のセキュリティと完全性が、組織のエンタープライズ・グリッドのセキュリティ・ポリシーに基づいて検証されなければならない。これには、コンポーネントから成る集合だけでなく、個々のコンポーネントも含まれる。
- プロビジョニング済みリソースの保存対象情報の完全性も検証されなければならない。これには、ログ・ファイルや暗号鍵データなど、デコミッションングが行われる前にリソースから収集されるデータが含まれる。
- 可用性は、要件ということだけでなく、設計の中心となるものであり、エンタープライズ・グリッドの特徴となる利点である。したがって、エンタープライズ・グリッド、GME、グリッド・コンポーネントの可用性を維持すること（また、それらに対する DoS 攻撃から守ること）は、1つの要件である。

6.2 識別 (Identification)

セキュリティ・システムの基本構成要素の 1 つに、あらゆるものを一意に識別できることがある。エンタープライズ・グリッドでは、すべてのグリッド・コンポーネントとユーザー・コミュニティがその対象となる。特にグリッド・コンポーネントは、プロビジョニングとデコミッションングが繰り返されるライフサイクルを通して、自分固有のアイデンティティを保持しなければならない。あるいはグリッド・コンポーネントのプロビジョニングまたは再プロビジョニングのたびに新しいアイデンティティが作成される場合、そのようにして作成されるアイデンティティの履歴を監査やフォレンジックのために記録しなければならない (これはおそらく GME の役割となる)。また、GME は、グリッド・コンポーネントとアプリケーション/サービスが自分が通信している相手ができるように、一意的に識別されなければならない。

6.3 認証 (Authentication)、認可 (Authorization)、監査 (Auditing) - (AAA)

アクセス管理関連の要件である認証、許可、監査も、グリッド環境に適用される。これらの要件に関して、グリッド環境に固有の側面として以下のことがある。

- GME、グリッド・コンポーネント、アプリケーション/サービス間の安全な通信を確保するために、各通信エンティティは相互に認証できなければならない。
- グリッド・コンポーネントは、定義されたセキュリティ・ポリシーに従う他のグリッド・コンポーネントと通信できる。そのような場合、グリッド・コンポーネントは、認可されることによってはじめて他のグリッド・コンポーネントと通信できる。認可の形式は組織とエンタープライズ・グリッドがデプロイメントされる環境によって異なり、厳格な形をとる場合と緩やかな形をとる場合がある。
- エンタープライズ・グリッド環境の監査機能は、グリッド・コンポーネントの動的結合と、その潜在的に短いライフサイクルを追跡し解決できなければならない。監査データは、監査されたグリッド・コンポーネントの再プロビジョニングまたはデコミッションングが行われた後もその意味を解釈できるものでなければならない。
- GME は、非グリッド環境内の AAA サーバーと同等の一連の機能を、エンタープライズ・グリッドに提供しなければならない。これには、ポリシー・ベースの拡張可能で「強力な」認証メカニズム (たとえば SAML や X.509 など) のサポートと、対グリッド・リソースの役割ベースのアクセス制御のサポートが含まれる。

6.4 任務の分離、最小の特権

アクセス制御ポリシーの、任務の分離と最小の特権という 2 つの標準も、エンタープライズ・グリッドに適用される。エンタープライズ・グリッド固有の側面に焦点を当てる場合、これらの標準は、GME とそれに関連する管理者に適用できる。この任務の分離をサポートするために、たとえば新しい管理者の役割 (たとえば「グリッド管理者」) を定義するという方法も理にかなっている。新しい役割も含め、あらゆる役割は、セキュリティのベスト・プラクティスの一環として、最小の特権 (「need-to-know」と「need-to-have」) の原則に基づいて設定する必要がある。

6.5 縦深防御 (Defense in Depth)

セキュリティのもう1つの一般的な原則として、縦深防御がある。従来のネットワーク化されたシステムでの一例を挙げると、ネットワーク・セグメントを非武装地帯 (DMZ) に物理的に分離する構成にするという方法がある。グリッド・コンポーネントのプール内の論理メカニズムを使用してそれを実現するとしても、エンタープライズ・グリッドでも DMZ などの従来の縦深防御の方策を維持する必要がある。縦深防御のための他の方策として、EGA 参照モデルの有向非巡回グラフ (DAG) を調べて、可能な場所でグラフの層ごとにシステム・セキュリティを強化するといったセキュリティ対策を講じることも可能である。

6.6 フェイル・セキユア

適切に設計されたどの情報セキュリティ・システムでもそうであるように、個々のグリッド・コンポーネント、GME、エンタープライズ・グリッドが、全体としてフェイル・セキユアとなるように設計されなければならない。エンタープライズ・グリッドはグリッド・コンポーネントのライフサイクルを非常に重視するので、要求された状態変化の結果は、成功となる場合もあれば失敗になる場合もある。フェイル・セキユアとは、グリッド・コンポーネントなどのエンタープライズ・グリッドの構成要素が、どんな種類のセキュリティ脅威に対しても、その脅威を受けやすい弱い状態にならないことである。

6.7 グリッド・ライフサイクルのセキュリティ要件

エンタープライズ・グリッド環境に固有のセキュリティ要件として、グリッド・コンポーネントのライフサイクルに関連するものもある。エンタープライズ・グリッド環境ではグリッド・コンポーネントが頻繁に再利用されるので、「安全なパッケージ化」、「安全な更新」、「安全なアーカイブ」、「安全な再利用」の機能があることが重要である。

- 「安全なパッケージ化」とは、グリッド・コンポーネントのグリッド内リソースへのプロビジョニングとグリッド内リソースからのプロビジョニング解除を容易に行えるように、グリッド・コンポーネントを論理的にパッケージできることである。これによって、各グリッド・コンポーネントをその他のコンポーネントから論理的に分離することが可能になる。さらに、これによって、変更管理、改訂管理、健全性管理をパッケージ単位で行える。同様に、サイトのセキュリティのポリシーと要件によっては、パッケージごとにデジタル署名および/または暗号化を行える。
- 「安全な更新」とは、デプロイメント済みのグリッド・コンポーネントを、より新しいバージョンのオブジェクトを使用して安全に更新できることである。これには、コンポーネントと安全に通信できることが含まれる。その目的は、既存の状態の問い合わせ、更新、および問題が検出されたときにグリッド・コンポーネントが前のバージョンにロールバックされるようにするためのチェックポイントの変更である。この機能は、既知の欠陥や弱点の修正、セキュリティのぜい弱性の修復、あるいは新しい機能の組み込みのために必要である。さらに、整合性を保とうという試みの一部としてどのグリッド・コンポーネントが更新を必要としているかを判別する際にも、この機能が使用される。このプロセスは、適切な変更管理と改訂管理を維持するうえで、「安全なパッケージ化」の概念を必要とする。

- 「安全なアーカイブ」とは、プロビジョニング済みのリソースから、後で必要になる可能性のある情報を抽出できることである。これには、前述のようなパッケージのほか、監査ログや暗号鍵データ、そして、保存されて資産の目的再割り当てやデコミッションングが行われても存続する必要がある機密性の高い他の構成やデータが含まれる。
- 「安全な再利用」は、EEPROM、メモリー、ディスク・スペースのデータ消去といった単純なことを指すが、より複雑なことを指す場合もある。この要件は、グリッドにとって特に目新しいものではない。たとえば、オペレーティング・システムはメモリーとディスクを使用する際に常にこれを行う。グリッド環境では、プールされ再利用されるあらゆるグリッド・コンポーネントに同じ要件が適用される。グリッド・コンポーネントは、次のユーザーによってアクセスされる前に、再初期化されなければならない（つまり新たに出直さなければならない）。ある状況でグリッド・コンポーネントの一部が複数のユーザーによって使用される場合、その部分はこの要件を持たす必要がある。

6.8 相互運用可能なセキュリティ

エンタープライズ・グリッド環境では、それが1つの同種の環境で構成されていることを前提にはできない。実際、「レガシー・システム」がエンタープライズ・グリッドの一部を構成する場合がある。したがって、ヘテロジニアスなグリッド・コンポーネント間で相互に運用可能なセキュリティをサポートすることが1つの要件になる。これは、GMEによって管理されるグリッド・リソースのプールに適用される。GMEは、そのようなプール内のすべてのリソースのセキュリティ属性を一律に管理できなければならない。また、エンタープライズ・グリッド全体に適用され機能する認証と認可のフレームワークをサポートするためには、ヘテロジニアスなグリッド・コンポーネントによって使用される様々なセキュリティ・モデルを相互にマップできなければならない。相互運用性は1つの制約とはなるが、ベンダー固有または製品固有の機能をグリッド・フレームワーク内で活用できる場合は、その使用を排除すべきでない。目的は、最小公約数でグリッド環境を構築することを強いることにあるのではなく、エンタープライズ・グリッド環境に製品と技術を容易に統合できるようにすることにあるからである。

6.9 安全な分離

多くの場合、エンタープライズ・グリッドの狙いは、複数のアプリケーションやサービスのリソースの必要性をグリッド・コンポーネントの共有可能プールを使用して満たすことにあるので、物理的、電氣的、あるいは論理的に分離されたサイロ型環境でもともと満たされている安全な分離に関する要件を満たすことは重要である。グリッド・リソースごとにどのようなタイプの分離が必要かが企業のセキュリティ・ポリシーで明示される場合がある（物理的分離、電氣的分離、論理的分離、分離不要など）。したがって、エンタープライズ・グリッドは、様々な企業独自のポリシーにも対応できるだけの柔軟なものでなければならない。分離の要件は、個々のグリッド・コンポーネントに適用されるだけでなく、グリッド・コンポーネントの集合体や、それを基に構築される高位のサービスにも適用される。

6.10 信頼関係

どのセキュリティ・インフラストラクチャ・システムでもそうであるように、エンタープライズ・グリッドを安全なものにするためには、ある特定の信頼関係をエンタープライズ・グリッドに設定する必要がある。これには、ユーザー、管理者、アプリケーション、サービスの GME および各グリッド・コンポーネントに対する関係が含まれる。これらの信頼関係は、根本的なセキュリティ・メカニズム（認証、または安全な通信チャンネルを持つことなど）によってサポートされる。また、これらは、GME とグリッド・コンポーネントによって追跡されなければならない。この例を以下に示す。

- コンピュータ、ネットワーク、ストレージといったリソースのどれが相互に連携できるか。
- 信頼がグリッド内でどのように確立、維持され、終了するか。
- 関連または依存するグリッド・コンポーネント間の信頼関係はどうか。
- 信頼の違反はどのように検出されるか。そのような違反が見つかったときに何が行われるか。

7 要約

エンタープライズ・グリッド環境には、ビジネス上の様々な魅力的なメリットがある。この環境に関連した固有の脅威とセキュリティ要件を理解することによって、リスクにさらされることを最小限に抑えてこれらのメリットを享受できる。さらに、これらのメリットの中のいくつかには、特に可用性の領域での機能強化されたセキュリティが含まれる。グリッド管理エンティティが人的リソース、プロセス、技術の組み合わせから、より自動化されたものに移行したとき、一元化されたセキュリティ管理の様々なメリットも実感できる。

8 関連活動

8.1 標準関連活動

- OASIS Security. <http://www.oasis-open.org/>
- OASIS Web Services Security TC. <http://www.oasis-open.org/>
- OASIS eXtensible Access Control Markup Language (XACML) TC. <http://www.oasis-open.org/>
- OASIS Security Services (Security Assertion Markup Language [SAML]) TC. <http://www.oasis-open.org/>
- Liberty Alliance Project. <http://www.projectliberty.org/>
- Global Grid Forum (GGF) Grid Security WG. <http://www.ggf.org/>
- OGSA WG Security Sub-Team. <http://www.globus.org/ogsa/>
- OGSA Security (Open Grid Service Architecture Security) WG. <http://www.globus.org/ogsa/>
- OGSA Authorization (Open Grid Service Architecture Authorization) WG. <http://www.globus.org/ogsa/>
- OGSA AuthZ WG. <http://www.globus.org/ogsa/>
- DMTF User and Security WG. <http://www.dmtf.org/>
- SNIA Management Protocol and Security TWG. <http://www.snia.org/>
- Trusted Computing Group (TCG). <https://www.trustedcomputinggroup.org/>

9 参照情報

- 「Enterprise Grid Alliance 参照モデル v1.0」 (2005年4月13日、EGA)
- 「企業におけるグリッド・ソリューションの採用の加速」 (2004年、EGA)

Enterprise Grid Alliance
2400 Camino Ramon, Suite 375
San Ramon, CA 94583
Tel: +1.925.275.6644
Fax: +1.925.275.6691
<http://www.gridalliance.org>