IoT時代の脅威分析とリスク評価

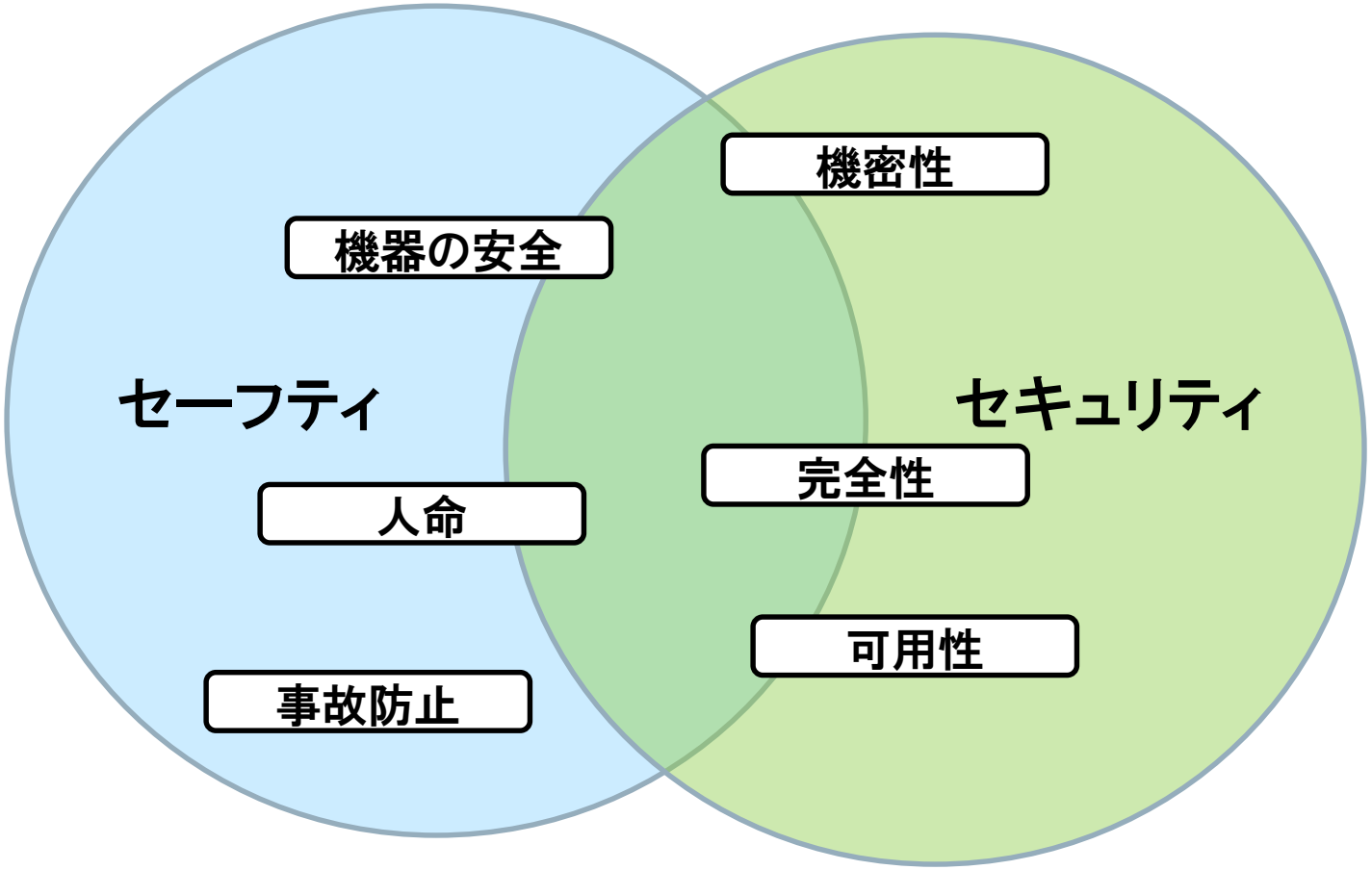
2017/11/17

情報セキュリティ大学院大学
大久保 隆夫

IOTを安全にするには

設計から安全にする

- IoTシステムでは個別の機器よりも全体を俯瞰した対策が重要
- 場当たり的なパッチでは限界
  →設計から安全にする=セキュリティ・バイ・デザイン
- セキュリティ、セーフティ双方の品質を確保
  - 手法として統一できるかは道半ば


機密性
機器の安全
セーフティ
セキュリティ
完全性
人命
可用性
事故防止

セーフティとセキュリティ
効果と要因から

効果	要因	規格、分析手法
セーフティ	セーフティ	IEC 61508 etc
	セキュリティ	なし
セキュリティ	セーフティ	なし
	セキュリティ	脅威モデリング etc

セキュリティ要求分析

セキュリティ要求分析

- 2つのアプローチ
  1. 攻撃者のゴールベースの分析
     攻撃者は何を狙っているか？
  2. 被害ベースの分析
     このシステムで何をされると困るか

1と2は包含関係であり、1、2を独立で(あるいは一方のみ)行っても、1→2あるいは2→1の順で行ってもよい

攻撃者の目標(1)

- STIX(Structured Threat Information Expression) version 1.2のCampaign(作戦)→Intended Effect(意図)参考
  情報処理推進機構:脅威情報構造化記述STIX概説
  https://www.ipa.go.jp/security/vuln/STIX.html
  ※STIX2.0では、Campaign→Objective(目標)
  https://oasis-open.github.io/cti-documentation/stix/intro

攻撃者の目標(2)

- 優位に立つ
  - 経済的に
  - 軍事的に
  - 政治的に
- 盗む
  - 知的財産
  - 認証情報
  - 識別情報
- アカウントのっとり
- 信用へのダメージ
- 競争で優位
- サービス品質の劣化
- 否認、欺瞞
- 破壊
- 名誉毀損
- 暴露
- 恐喝
- 詐欺
- ハラスメント
- 制御システムの操作
- 抜け道
- 認可されていないアクセス

被害

- 信用の喪失
- 競争優位の喪失
  - 経済的
  - 軍事的
  - 政治的
- データの破棄、破壊
- サービス品質の劣化
- 破壊
- サービス、操作の中断
- 経済的損失
- 機密、知的財産権情報、専有の喪失
- 意図しないアクセス
- 利用者のデータの喪失

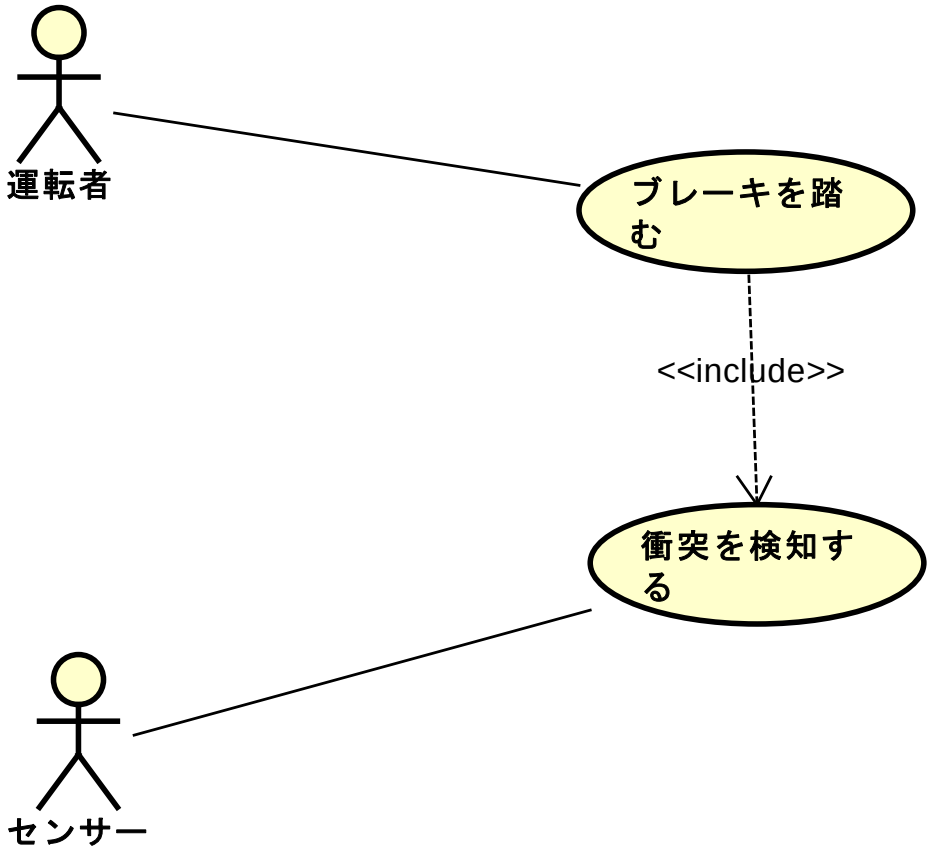
資産の識別

■ 資産の識別

1. ユースケースの作成
2. ユースケースシナリオから資産候補の抽出
3. 資産候補が攻撃者の目標or被害に合致するか検証→資産として抽出

資産識別の例(1)
自動ブレーキ

- 自動ブレーキの例



資産識別の例(2)

■ シナリオ1

1. 利用者がブレーキペダルを踏む
2. ブレーキ信号をコントローラに伝える
3. ブレーキが作動して止まる

■ シナリオ2

1. センサーが衝突を検知
2. 衝突可能性情報をブレーキのコントローラに伝える
3. コントローラが自動ブレーキを作動させる

資産識別の例(3)

資産候補

- コントローラ
- ブレーキ
- ブレーキ信号
- 衝突可能性情報

攻撃者の目標とつきあわせ→資産抽出

- ブレーキ+制御の不正な操作
  =危険！
- ブレーキは資産として抽出
- 同時に、「ブレーキに対する不正操作」を脅威として識別

更なる脅威抽出

- ガイドワードの適用
  - 資産に対してか？ユースケースに対してか？
    シナリオに対してか？
- エンティティへのSTRIDE適用→攻撃者目標を達成するか
  - なりすまし
  - エンティティの改ざん

ガイドワード候補

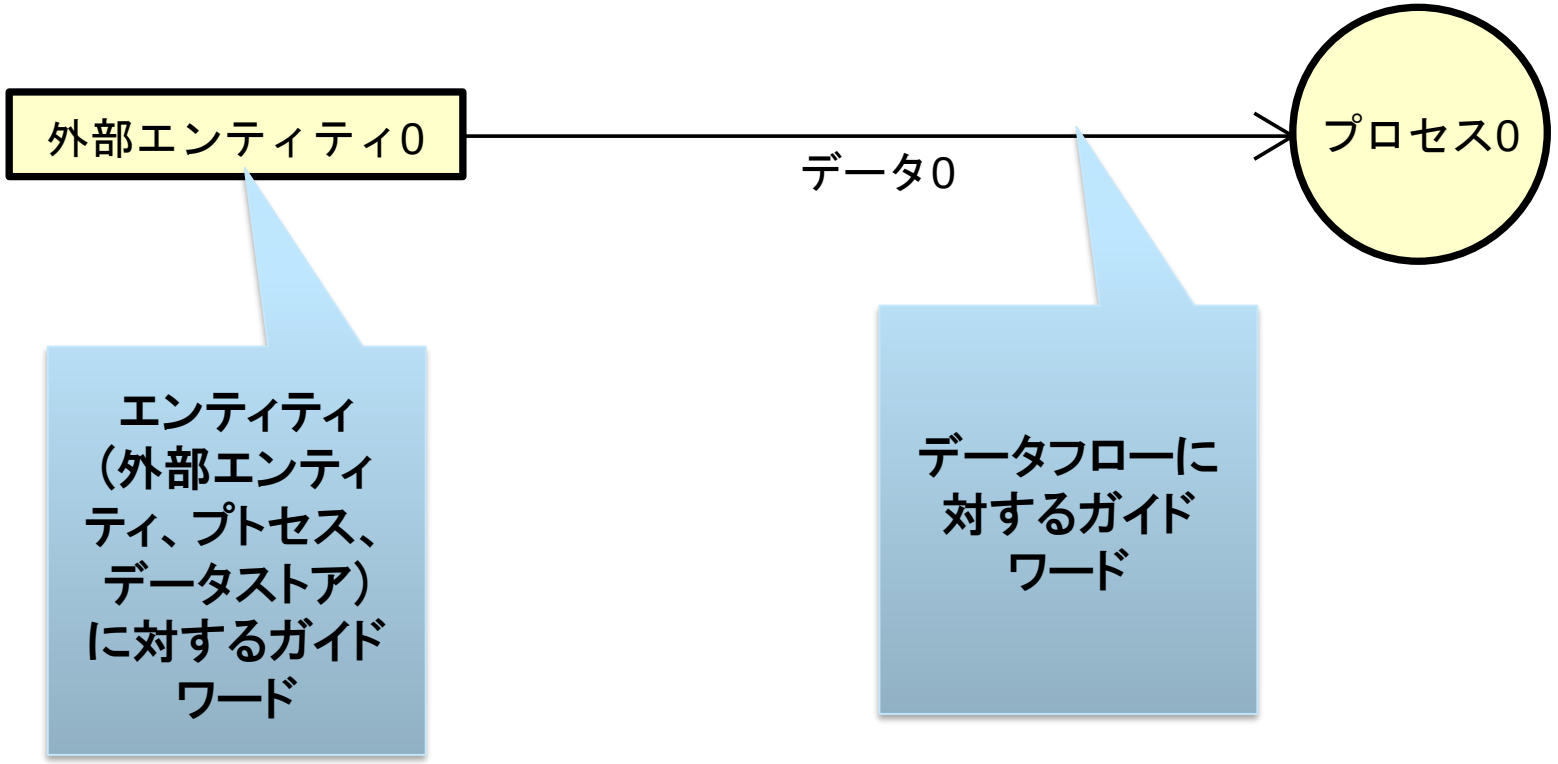
- STRIDE
- HaZopのガイドワード
- STAMP/STPAのガイドワード

STRIDE

エントリポイントにおいて脅威を発見するための
ヒント、脅威分類

- Spoofing(なりすまし)
- Tampering(改竄)
- Repudiation(否認)
- Information disclosure(情報の漏洩)
- Denial of service (DoS攻撃)
- Elevation of privilege(権限昇格)

STRIDEのガイドワードとしての役割



エンティティからの情報漏洩

フローからの情報漏洩

HazOp

- 特徴
  - システムの機能やプロセスに対し、潜在的に含まれたリスクを、体系的に分析
  - 「ガイドワード」の概念による効率的な分析
- 化学工業の分野で確立(1963年)
- 国際標準:IEC 61882

HaZopのガイドワード

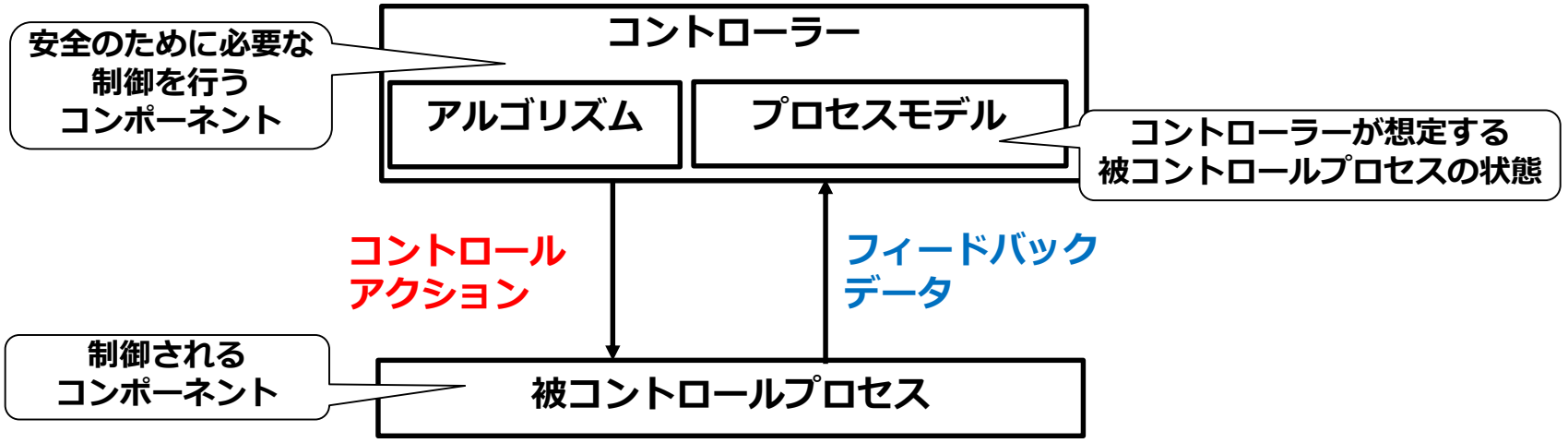
現象に対するガイドワード

ガイドワード	意味	例
No (not, none)	設計意図が全く達成されない	期待した製品の送出がない
More (more of, higher)	パラメータの定量的増加	設計より高温状態が発生
Less (less of, lower)	パラメータの定量的減少	設計より低圧状態が発生
As well as (more than)	追加の動作が発生	(故障や人的ミスにより)同時に別のバルブが作動
Part of	設計意図が一部しか達成されない	(全停止のはずが)システムの一部のみ停止
Reverse	設計意図と反対の状況が発生	システム停止時に逆流発生
Other than (other)	全く異なる動作が発生	気体用パイプに液体が流出

STAMP

STAMP（**S**ystems-**T**heoretic **A**ccident **M**odel and **P**rocess）：システム理論に基づく事故モデル

- 「要素（コンポーネント）」と「相互作用（コントロールアクション）」に着目してメカニズムを説明
- 「アクションが働かない原因」＝「コントロールアクションの不適切な作用」という視点を持つことで原因を有限化



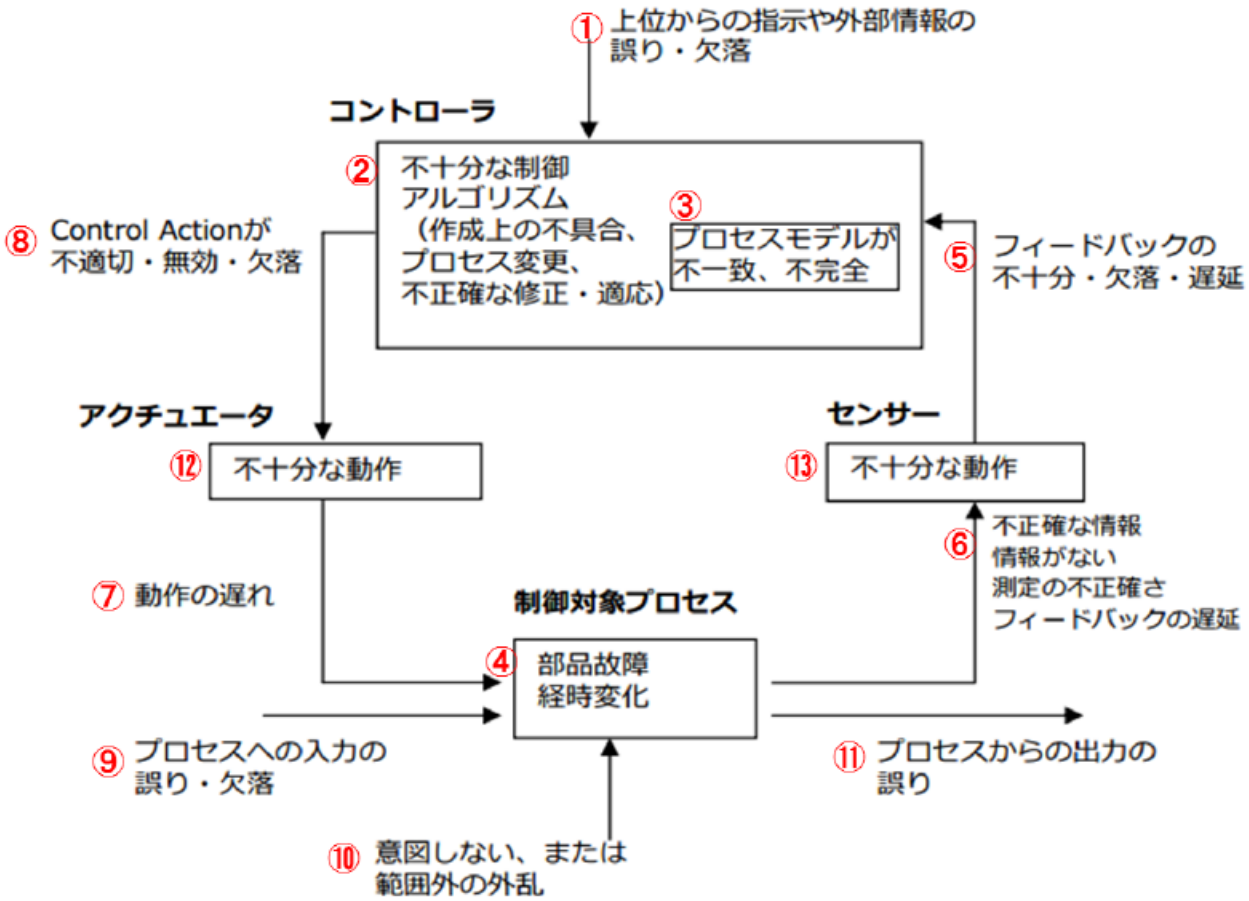
安全解析手法STPA

- STPA(STAMP based Process Analysis)STAMPに基づく安全解析手法
- MITのNancy G. Leveson教授が提唱
- 複数のコントローラが介在する複雑なシステム(Complicated system)に対する安全解析の方法論
- ガイドワードを使い、unsafe(非安全)なコントロールアクションを想定
- ヒントワードを使い要因を分析

STPAのガイドワード

- 与えられるとハザード
- 与えられないとハザード
- 早すぎる/遅すぎるとハザード
- 長すぎる/短かすぎるとハザード

STPAのヒントワード



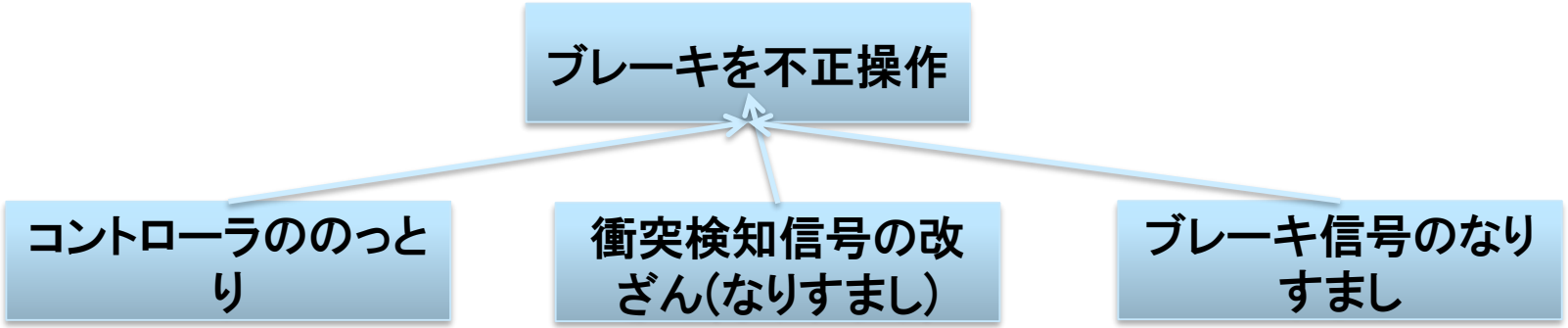
出典:IPA「はじめてのSTAMP/STPA(実践編)」

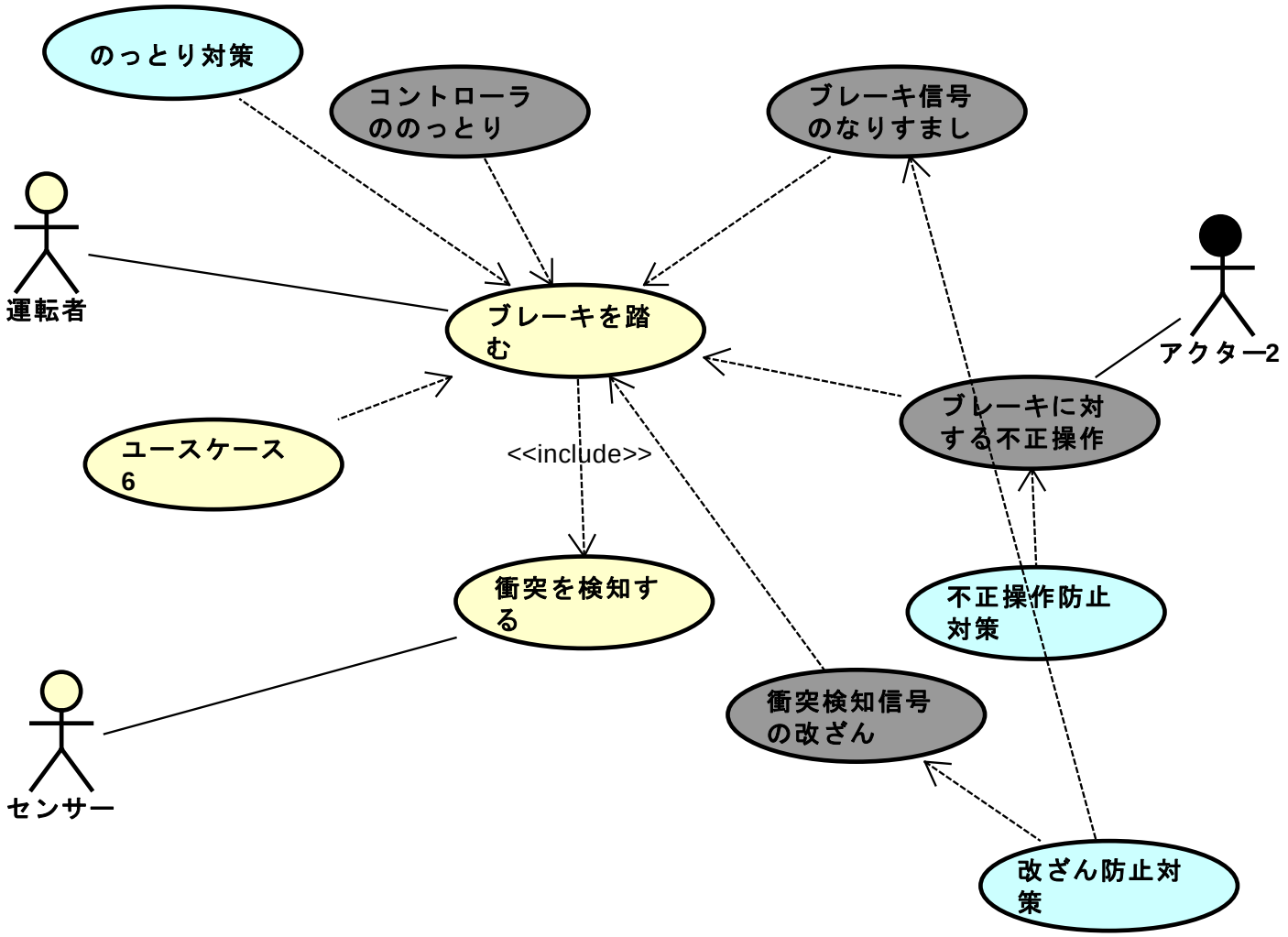
セーフティ系ガイドワードを脅威分析に活かすには

- HaZop、STPAのガイドワード:攻撃者の目標ないし被害事象に直結
  - No/More…の原因になるセキュリティ脅威を導出する
    - ex)改ざん(T)により、Nol(More)
  - 情報漏洩系は非対応
- STPAのヒントワード:セーフティ寄りで、セキュリティ脅威分析には不向き

脅威の詳細化

- 判明している情報のみで詳細化する



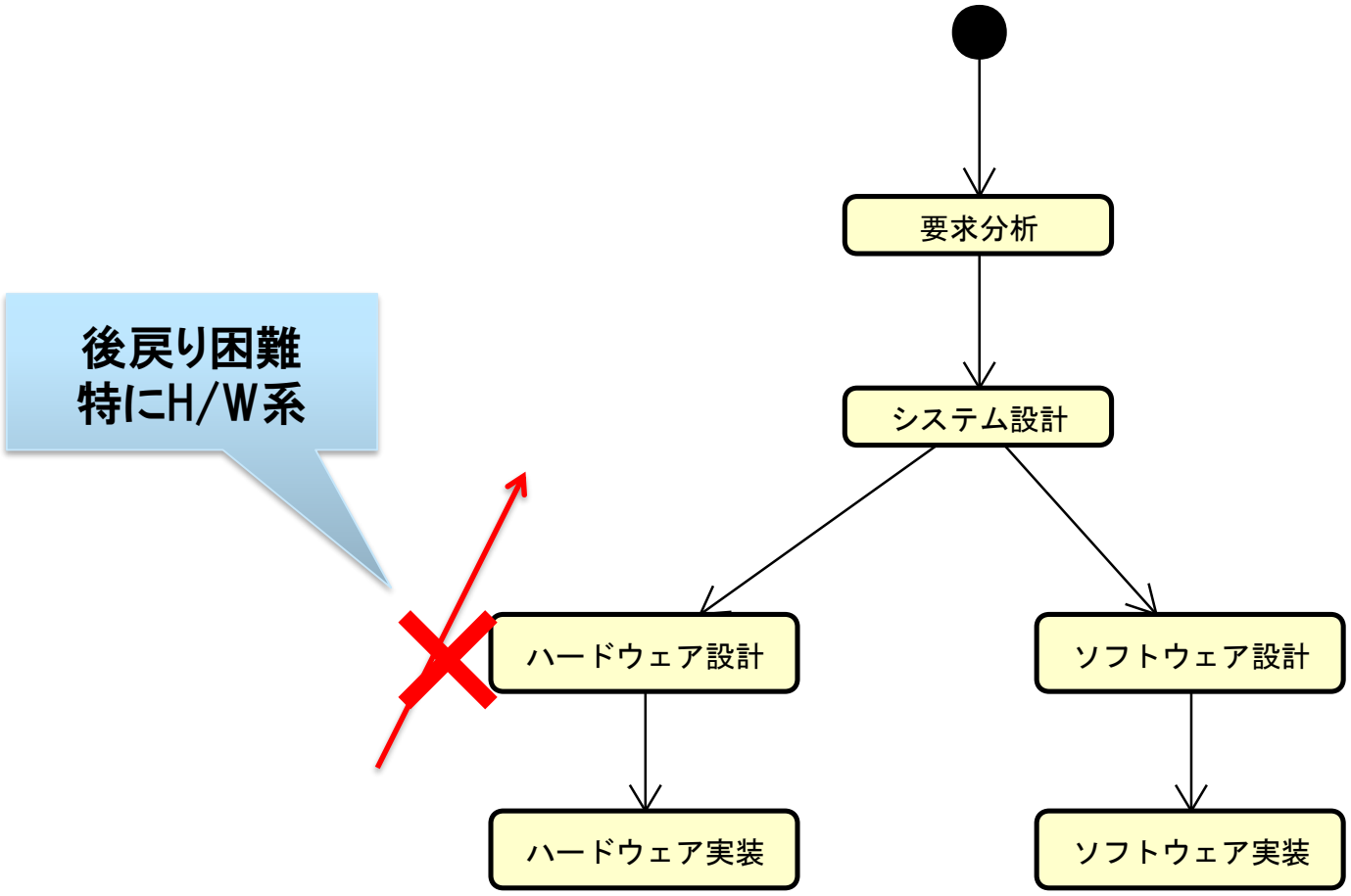

のっとり対策
コントローラののっとり
ブレーキ信号のなりすまし
運転者
ブレーキを踏む
アクター2
ユースケース6
<<include>>
ブレーキに対する不正操作
衝突を検知する
不正操作防止対策
センサー
衝突検知信号の改ざん
改ざん防止対策

脅威分析の前に行うべきこと
アーキテクチャのセキュリティ評価

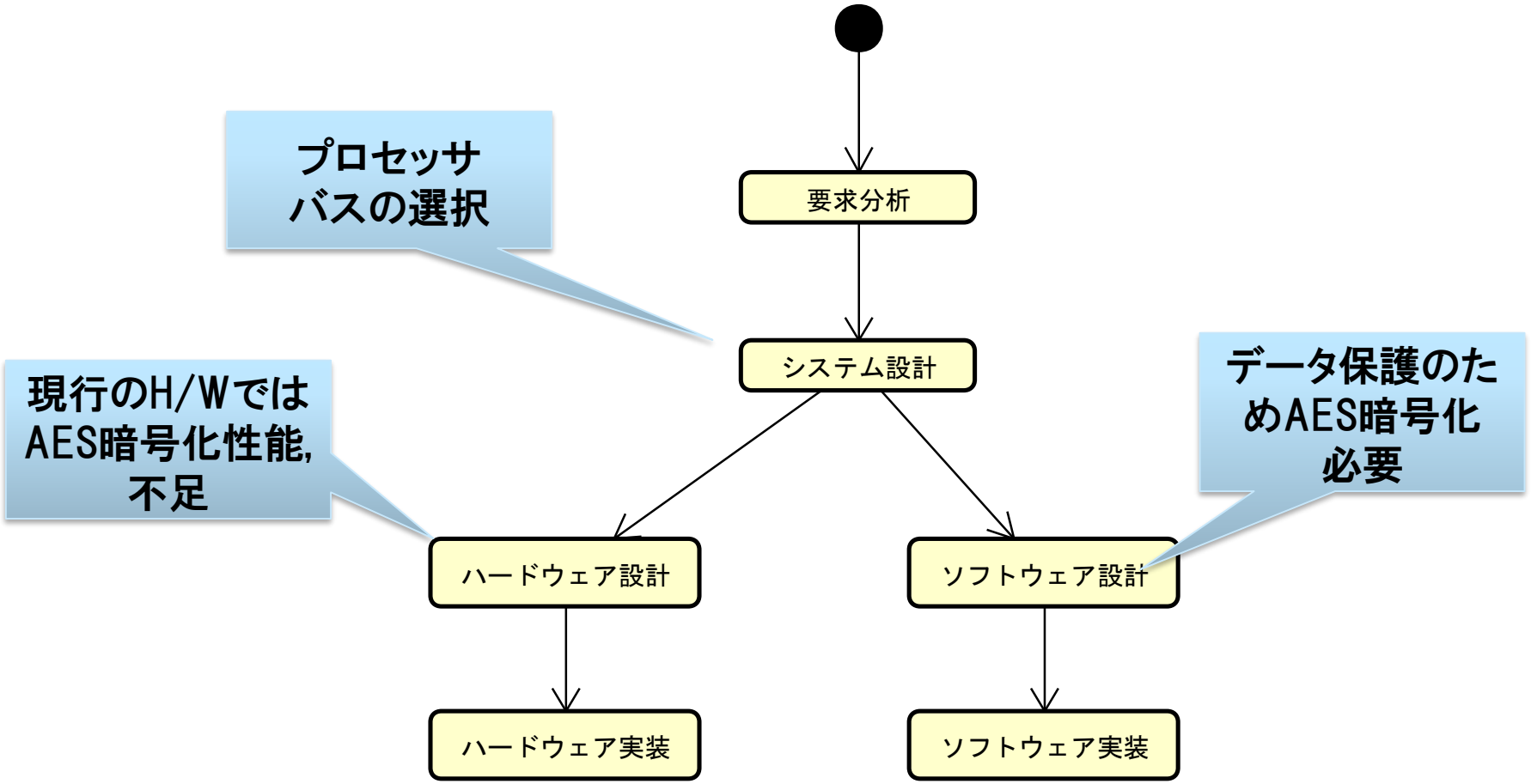
アーキテクチャなしの分析で十分か

- 通常のソフトウェア開発では、アーキテクチャの選定は設計時に行う
- しかし、誤ったアーキテクチャ選択はセキュリティに致命的な(対処不可能な)リスクを生む可能性がある

組込み系における開発


後戻り困難
特にH/W系
要求分析
システム設計
ハードウェア設計
ソフトウェア設計
ハードウェア実装
ソフトウェア実装

クリティカルな手戻りの例



手戻りを防ぐには

- 要求分析後、設計前にハードウェア、ソフトウェアのアーキテクチャ候補を列挙
- それぞれの候補についてリスク評価を行う
  - CVE、CWEなどの評価を参考
  - 明確な脆弱性がないか
  - データ保護、アクセス制御などを実現可能か

脅威分析

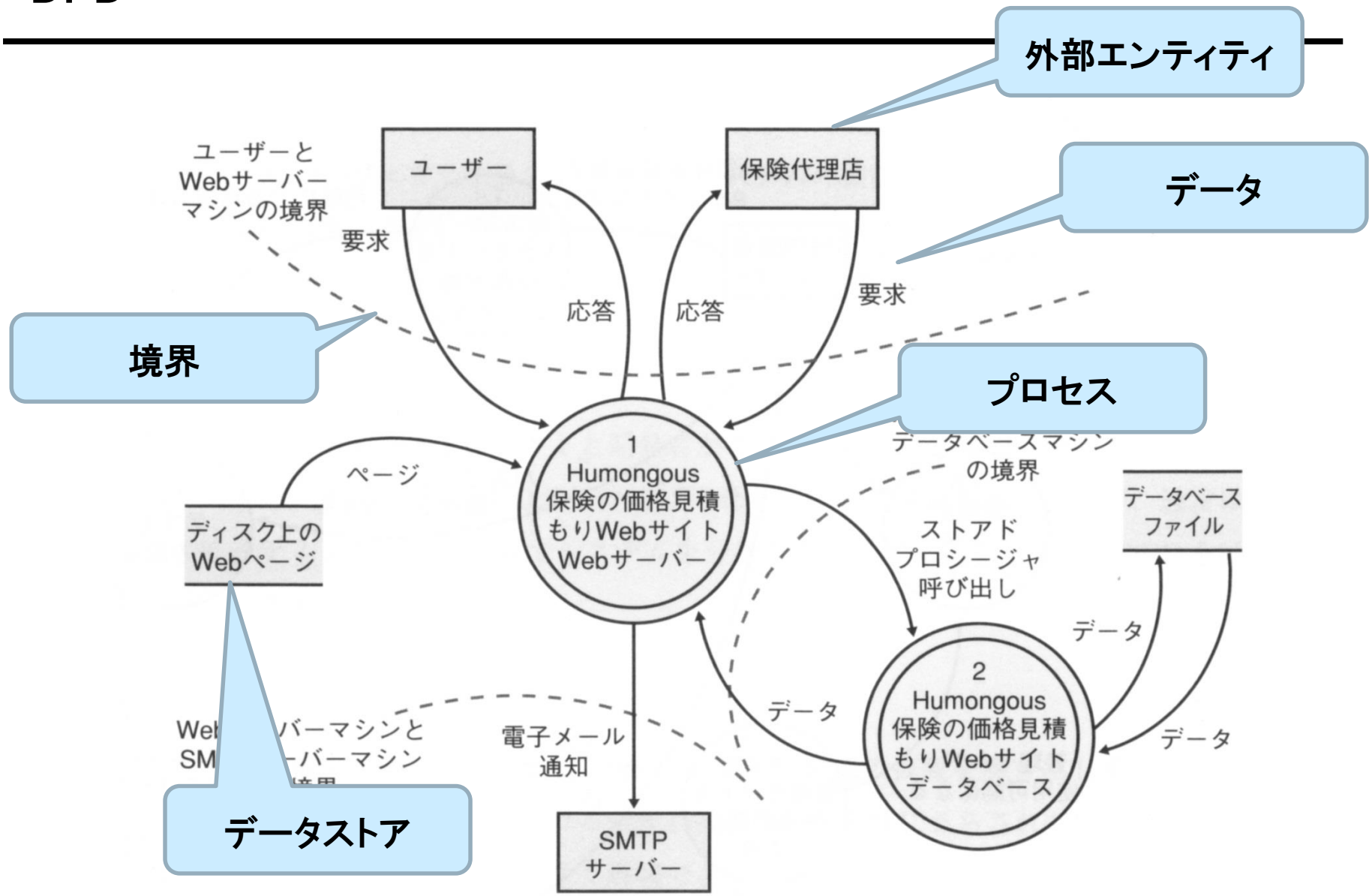
脅威分析

- 設計段階においては、資産ベースの分析よりも、モデルベースの分析を重視
  - 脅威モデリング
- セキュリティ要求を侵害しそうな脅威を識別し、対策する
- アーキテクチャ選択時に、想定済の典型的脅威を利用

脅威モデリング

- Microsoftが考案した脅威分析手法
  - 脅威分析の中では一般に最もよく使われている
- 設計したシステムにおける脅威分析(脅威の抽出、評価)を行う手法
  - Data Flow Diagram(DFD)を用いた脅威抽出
  - STRIDEによる脅威発見
  - アタックツリー等による脅威のリスク評価
- アーキテクチャが明確なとき、脅威抽出の手法としては有効
- 参考書
  - [HL04]M.Howard, D.LeBlanc: WRITING SECURE CODE, Microsoft press,2004.
  - [Swiderski05] Swiderski, F. and Snyder, W.: 脅威モデル ― セキュアなアプリケーション構築, 日経BPソフトプレス (2005).
  - [Sho14] A.Shostack: Threat Modeling , Wiley (2014).

DFD

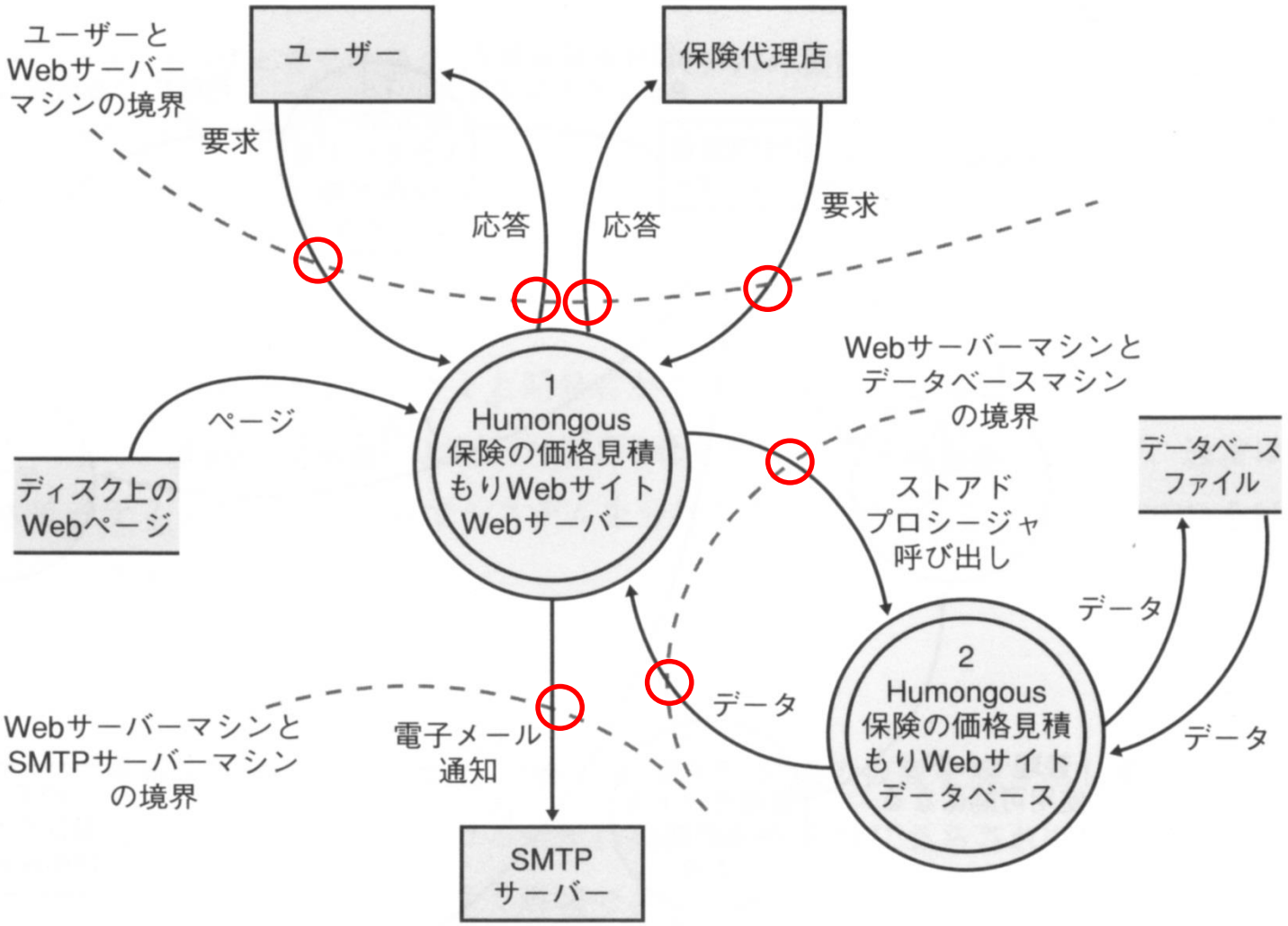


出典：M.Howard et al: “Writing Secure Code”, Microsoft press

DFD
境界の意味

- データ送受信の相手が信頼できるか？を考える
  - 物理的境界を越えてくるデータは信頼できない
  - 外部エンティティから来るデータは信頼できない
  - 信頼できない相手にデータを送るのは危険
- エントリポイント:脅威の起きそうなポイント
  - 境界とデータフローの交点
  - 境界を越えてデータがやりとりされる時に脅威が発生しやすい
  - 信頼できない相手から来たデータ:改ざんされているかも？
  - 信頼できない相手にデータを送る→流出するかも？

エントリポイント



出典：M.Howard et al: “Writing Secure Code”, Microsoft press

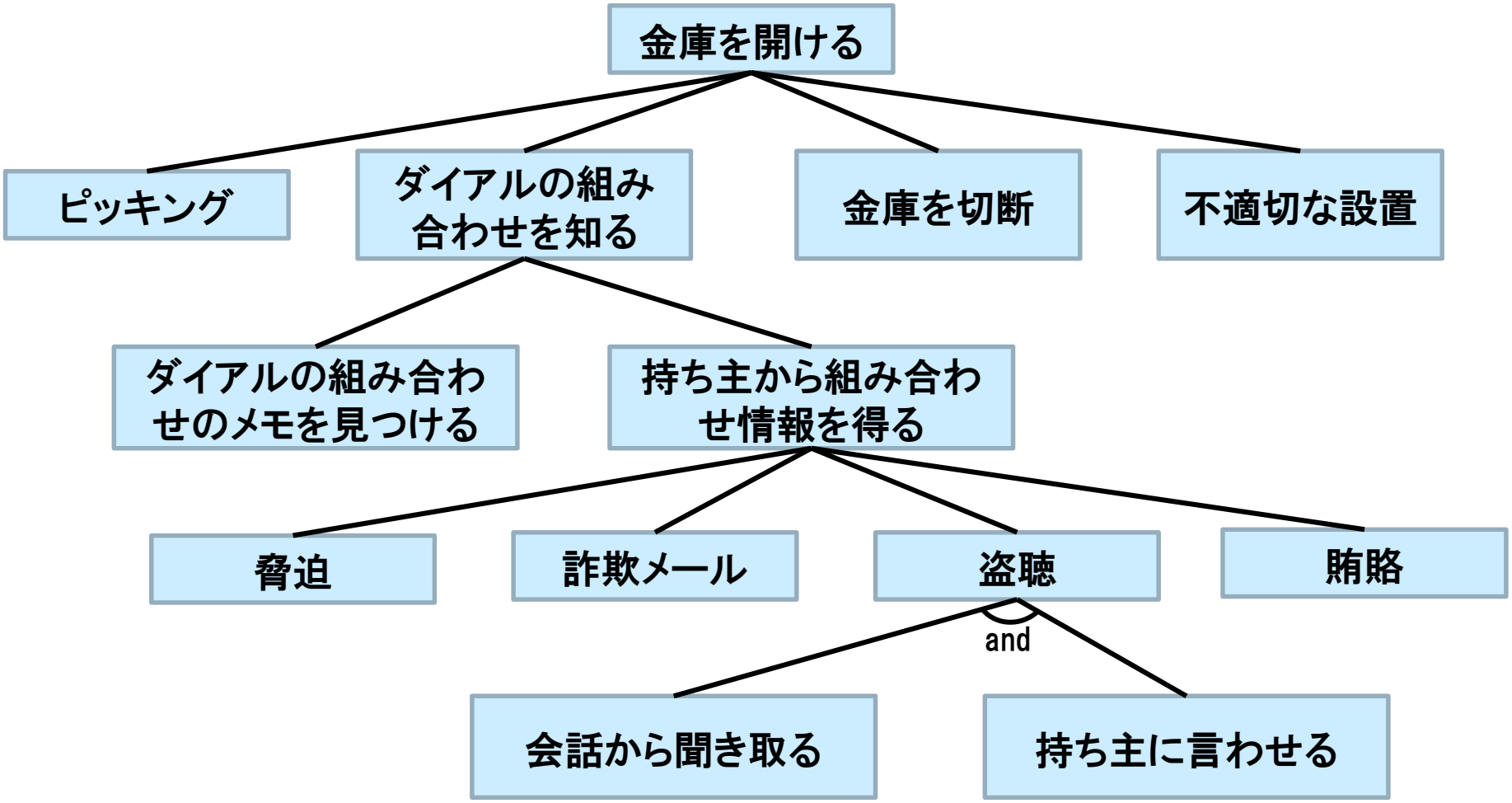
アタックツリー

- 脅威分析手法の一つ
- 脅威をTree状に詳細化していく
  - 具体的な攻撃手段とその可能性を明確化
  - ⇒対策の糸口
- 原典
  - [SCH99] B. Schneier, “Attack trees: modeling security threats,” Dr. Dobb’s Journal, December 1999.
  - [MEL01] B A. P. Moore, R. J. Ellison, and R. C. Linger, “Attack modeling for information security and survivability,” CMU/SEI-2001-TN-001, March 2001.

アタックツリー分析の手順

- 上位の攻撃を考える
- その攻撃を実現する手段や条件を下位ノードとして接続
  - 下位ノードが独立である場合:線で接続するのみ
  - 下位ノードの組み合わせで上位が実現できる場合:and関係で接続

アタックツリー
分析の手順



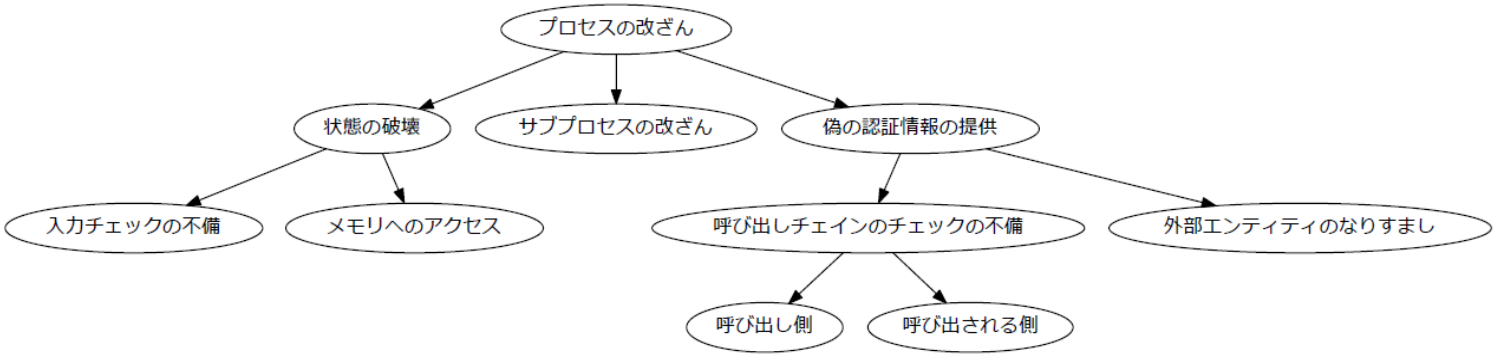
脅威詳細化に活用する知識

- Threat Tree Patterns
- CAPEC(Common Attack Pattern Enumeration and Classification)

Threat Tree Patterns

- Microsoftが提供するThreat Tree(アタックツリーと似ているが、脅威ベースで分解)のパターン
  - STRIDE以降の分解がされているので、ある程度までは活用できる
  - 個別の詳細設計までのブレークダウンには非対応

Threat Tree Patternsの例



CAPEC

- CAPEC(Common Attack Pattern Enumeration and Classification)
- 攻撃パターンのデータベース
- https://capec.mitre.org/
- いくつかの観点での階層構造があるが、アタックツリー分析にはあまり適していない

CAPECの利用法

- CAPECの各パターンには、「Motivation(攻撃者のモティベーション)」というパラメータがある
  - 攻撃者の目標から該当するものを検索
- アーキテクチャからの特定
  「Abstract」や、「Resources Required」から