ブロックチェーンを用いたデジタルコンテンツの流通

神奈川大学
201603790
松尾 浩貴

研究背景・問題点

- 近年、ネットワークの普及と発展と共に、
  デジタルコンテンツの流通が盛んになっている

- 現在普及しているDRM(デジタル著作権管理)ではデジタルコンテンツが他のデジタルコンテンツの利用を制限することやコンテンツ購入後のユーザ間の自由な取引ができない

- コンテンツとその利用者、コンテンツとそれを利用するコンテンツの関係を明確にしてデジタルコンテンツを保護する必要がある

神奈川大学　松尾 浩貴

提案

- 権利の移転を、主体を利用者とコンテンツ著作者、対象をデジタルコンテンツとしたTake – Grant Model で表現する
- 不正なコンテンツ利用をCovert channel を用いてモデル化することでコンテンツ著作者が意図しない権利の流通を分析し、不正な取引の防止を可能にする
- Covert channel の解析システムをブロックチェーン技術であるスマートコントラクトを用いて実行することで、利用者は取引実行の有無とコンテンツに対する権利を保障できる

神奈川大学　松尾 浩貴

Take-Grant Model

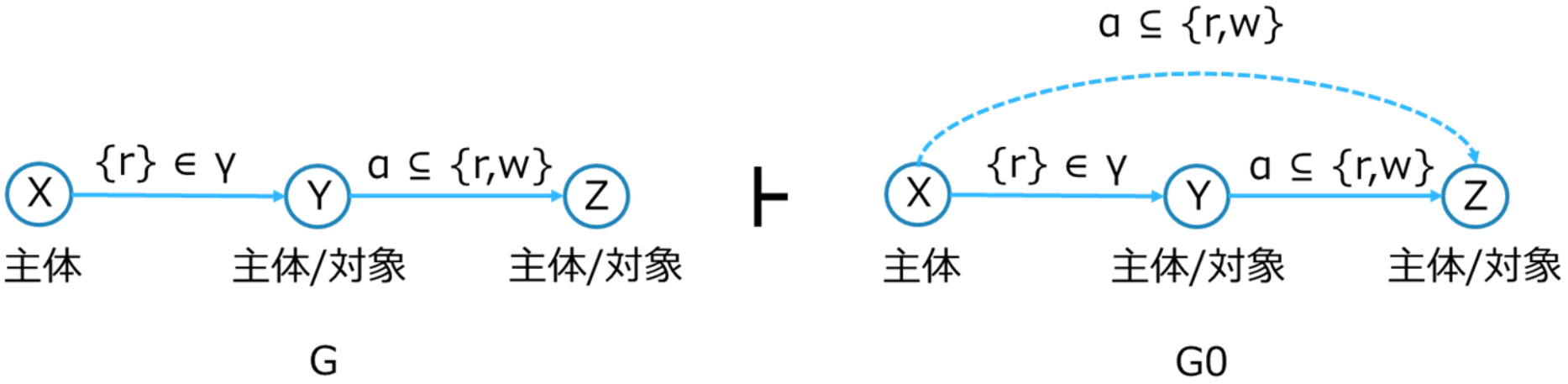
- 権限の委譲を記述することを特徴とする
  セキュリティモデル
- 権限の委譲をグラフ理論に基づいて分析する
- システムの保護状態は、主体および対象を頂点、権限を
  ラベル付きの辺とする有限の有向グラフによって表される
- 権限の委譲を行う操作はTake, Grant, Createの
  ３つである

神奈川大学　松尾 浩貴

Take-Grant Model

Take

r : Read, w : Write

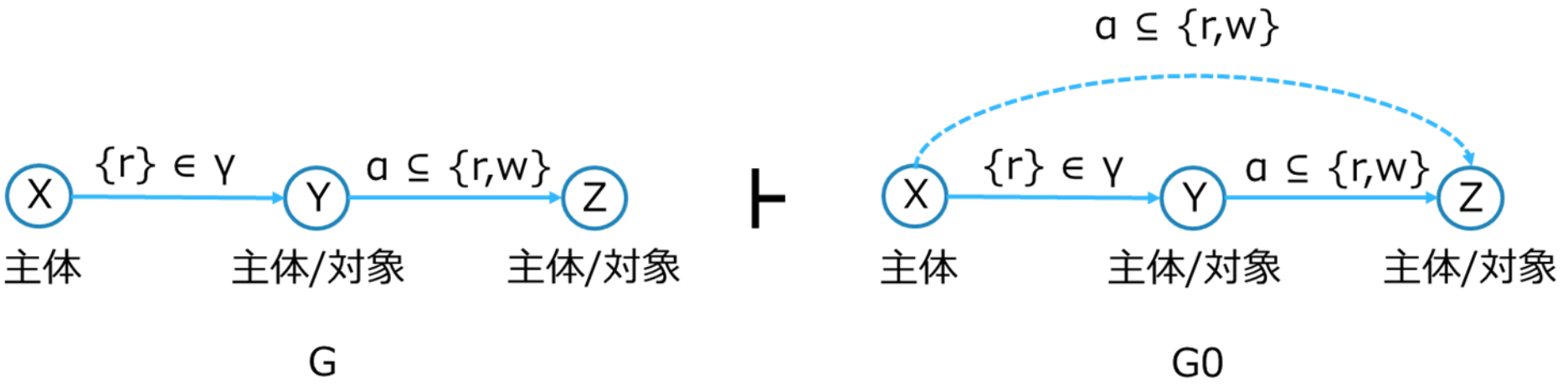


一定の条件のもとで、主体が他の主体 (または対象) に
権限を委譲される

Take-Grant Model

Take

r : Read, w : Write

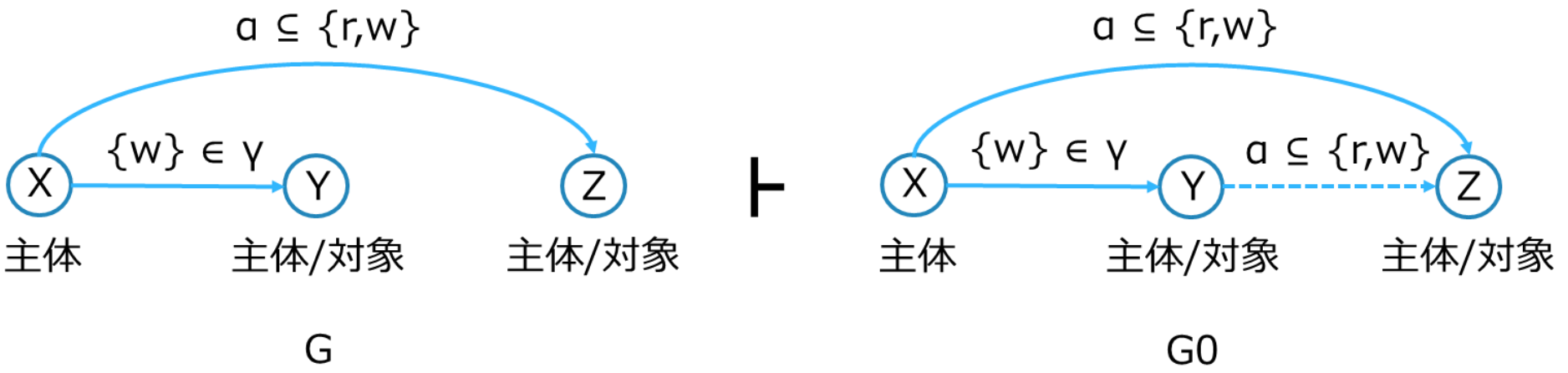


グラフの書き換えによって、ノードXはノードYから
ノードYの持つノードZへの権限を委譲されている

Take-Grant Model

Grant

r : Read, w : Write

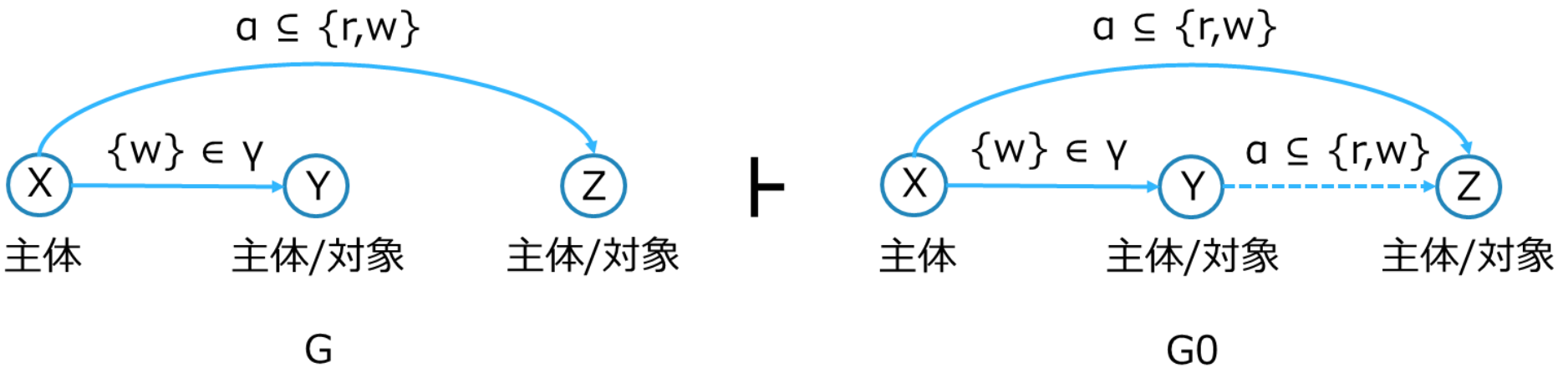


一定の条件のもとで、主体が他の主体 (または対象) に
権限を委譲する

Take-Grant Model

Grant

r : Read, w : Write

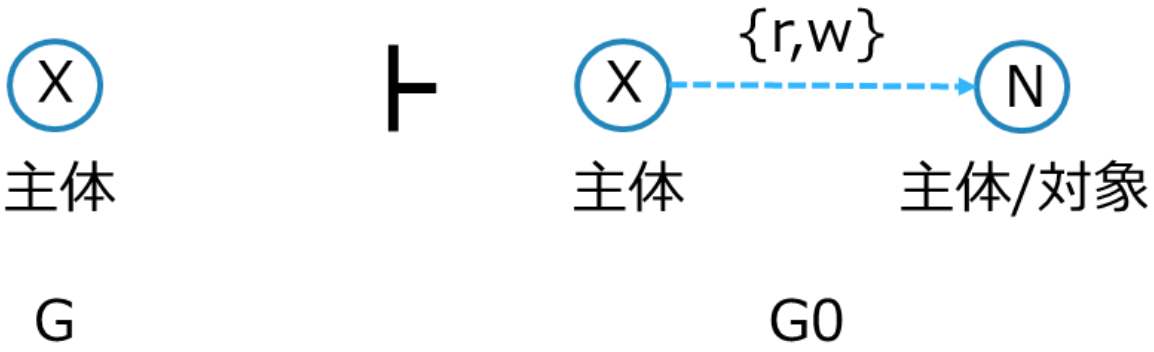


グラフの書き換えによって、ノードXはノードYに
ノードXが持つノードZへの権限を委譲している

Take-Grant Model

Create

r : Read, w : Write

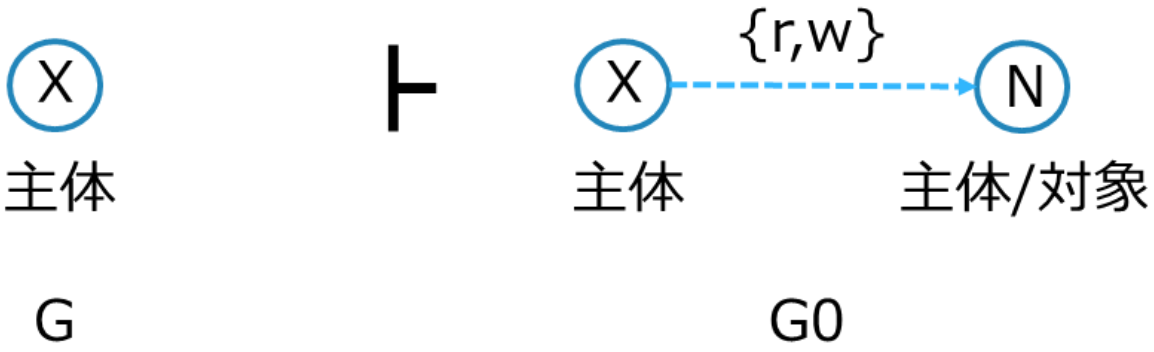


一定の条件のもとで、新たな主体 (または対象) を作成する

Take-Grant Model

Create

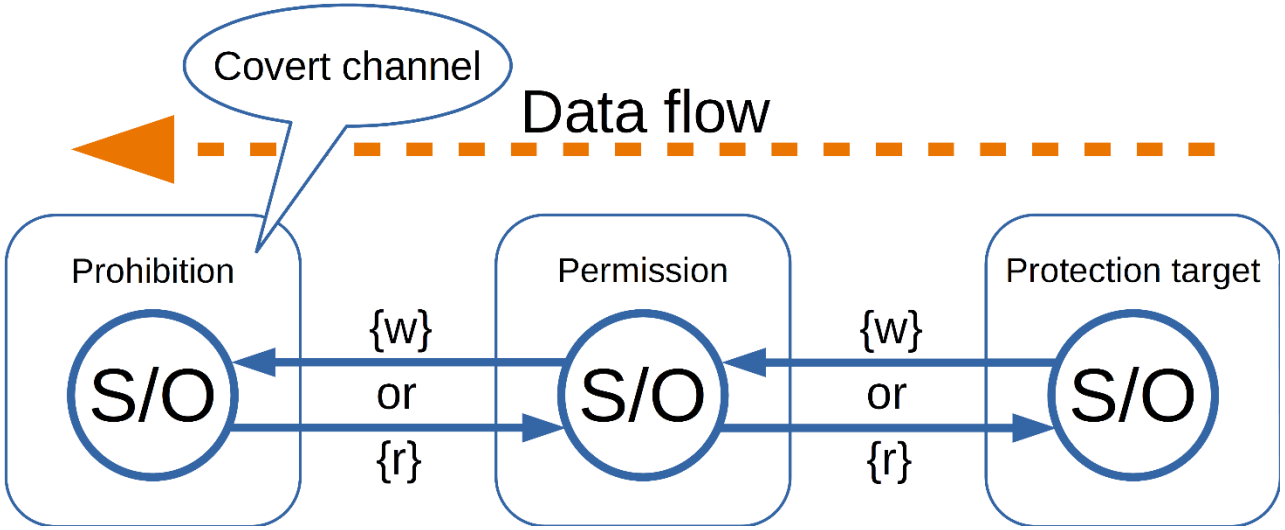
r：Read, w：Write



グラフの書き換えによって、ノードXは新たなノードNを生成し、ノードNへの権限を得る

Take – Grant Modelにおける Covert channel

- Take – Grant Modelにおいて、ノードが直接的に辺で繋がっていないにもかかわらず、多数のノードを通じて間接的にノードの情報を読むことをCovert channel （関節情報フロー）と呼ぶ
- 提案するDRMのモデルにおいて Covert channel の発生を不正なデジタルコンテンツの利用にモデル化する



神奈川大学　松尾 浩貴

DRMのためのTake-Grant Model

Covert channel 分析システムのアルゴリズムは、

- 権限移譲によってグラフの書き換えを検知
- 新たな辺の始点または終点のノードに保護対象ノードがあるかどうか検知する
- そのノードから Covert channel が発生する方向にBFS (幅優先探索) を行う
- 許可を受けていないノードがあるか探索、もしあれば Covert channel を分析する
- 探索過程で通ったノードに保護対象ノードの情報の保持を示すフラグを立てる

神奈川大学　松尾 浩貴

DRMのためのTake-Grant Model

BFSの探索過程における計算量削減方式

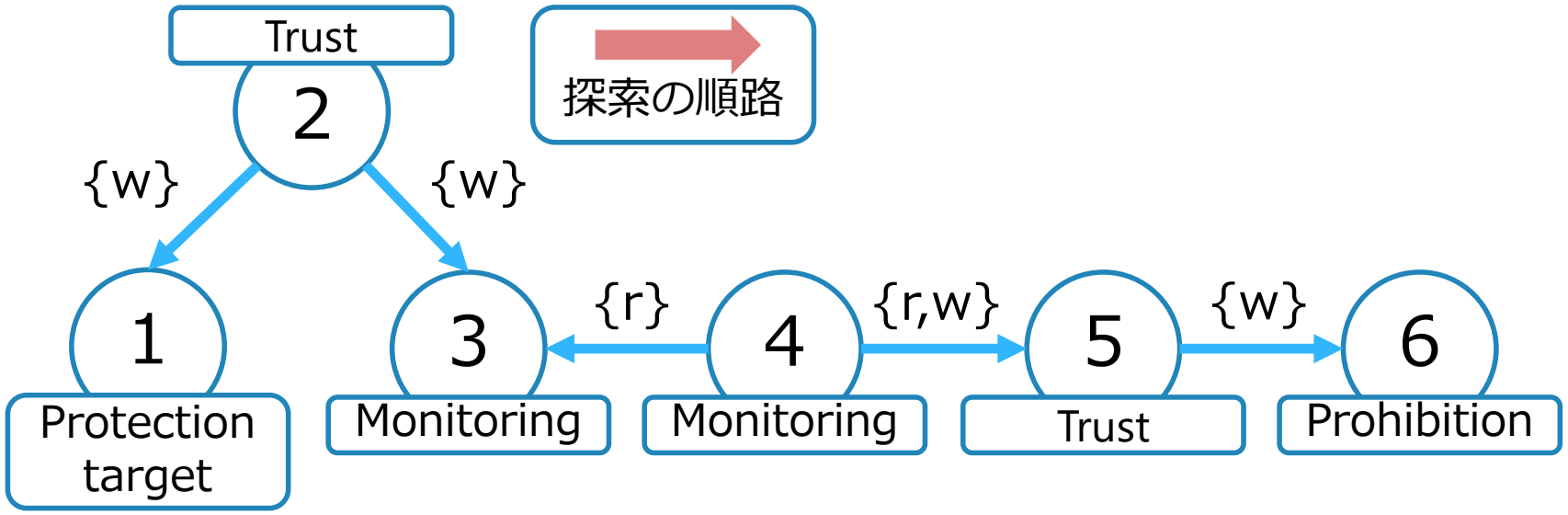
三つのノードの定義：

1. **信頼ノード**：保護対象ノードが課す他ノード に対する制限の状態が Trust (信頼) である、または ノードが保護対象ノードの情報をすでに保持しているならばそのノードには訪れない
2. **監視ノード**：保護対象ノードが課す他ノード に対する制限の状態が Monitoring (監視) であるならば、保持を認めるがそのノードの先の探索は行われる
3. **禁止ノード**：保護対象ノードが課す他ノードに対する制限の状態がProhibition (禁止) ならばただちにBFSの中断を行う

停止条件：最終的に中断されることなく、訪れる先のノードがなくなった場合、基本操作によって Covert channel が発生していない、つまり不正利用が発生していないことを示す

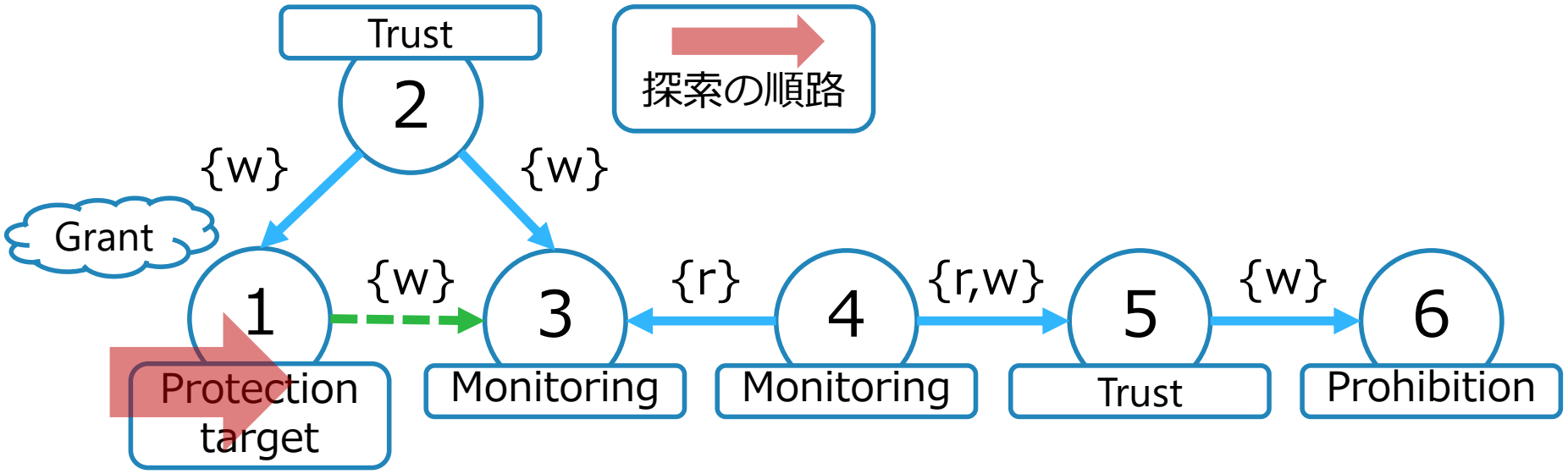
神奈川大学　松尾 浩貴

DRMのためのTake-Grant Model



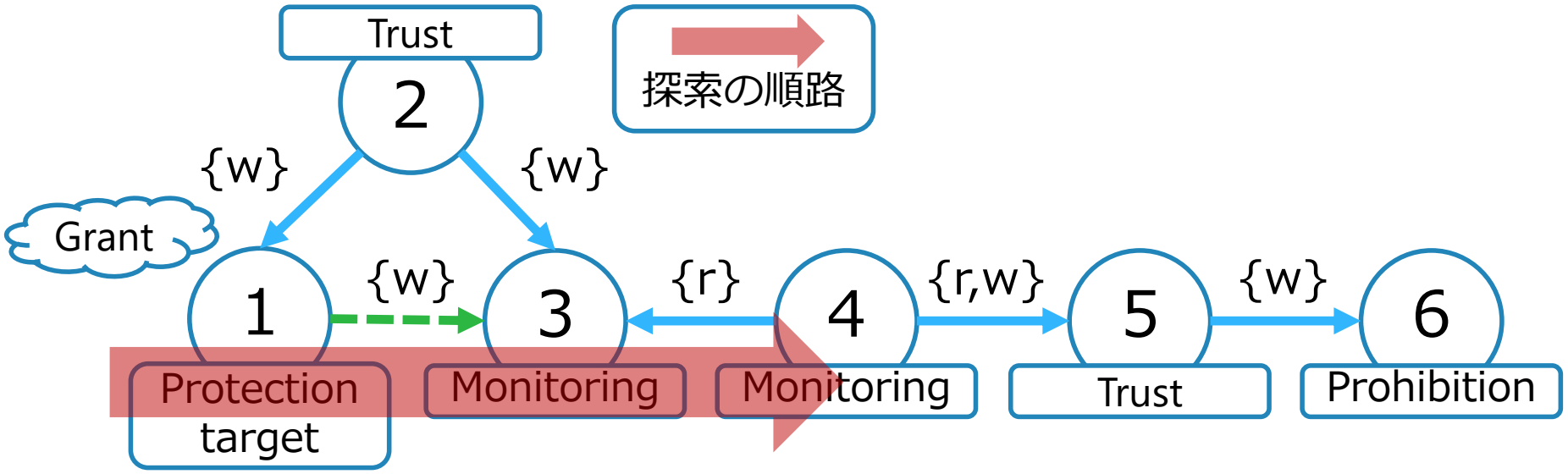
次のようなグラフの場合、ノードの状態がBFSの探索に
どのような変化を与えるか示す

DRMのためのTake-Grant Model



Grantによりノード1,3間に新たな辺ができた場合、
ノード1が保護対象ノードであるためノード1から探索開始

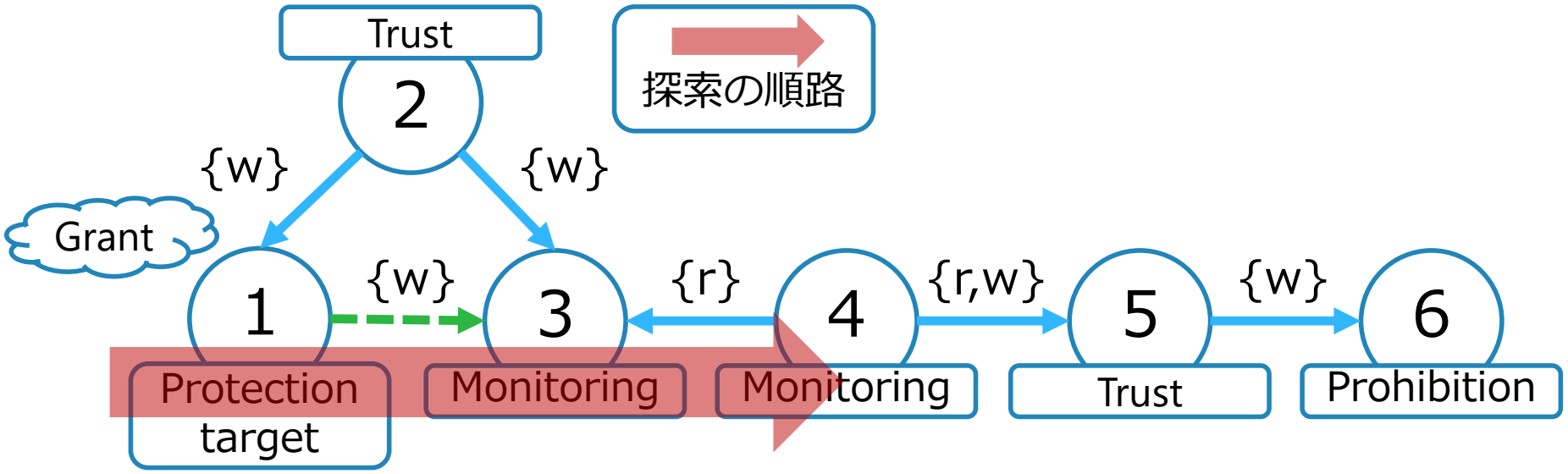
DRMのためのTake-Grant Model



監視ノードである3,4は探索経路となるため、探索を行う
信頼ノードである2,5は探索経路にならないため無視される

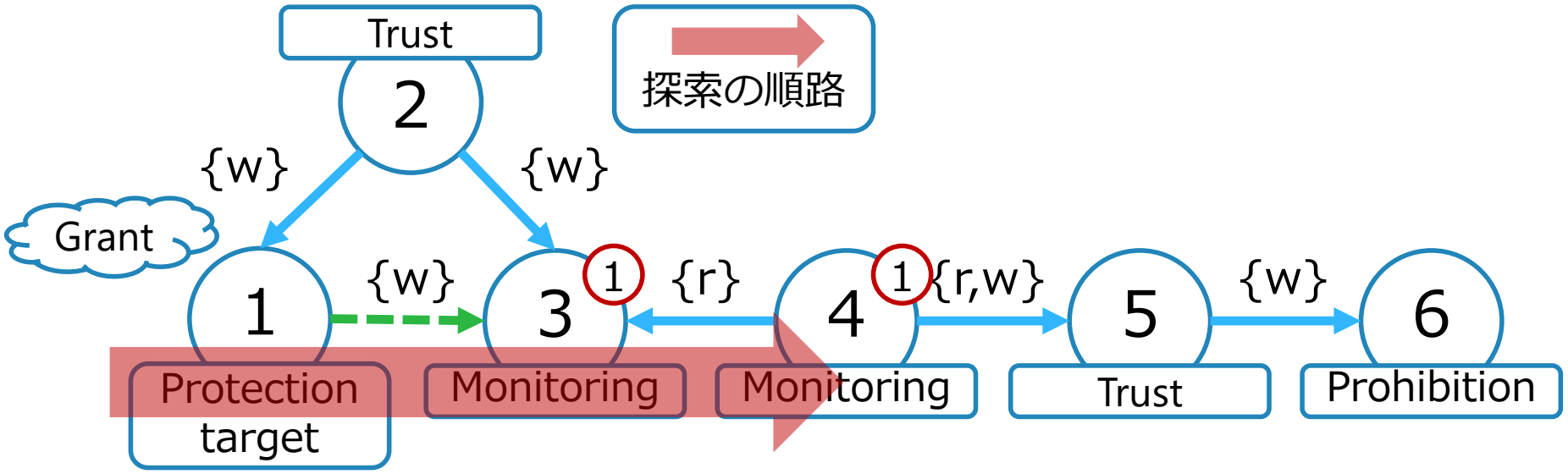
神奈川大学　松尾 浩貴

DRMのためのTake-Grant Model



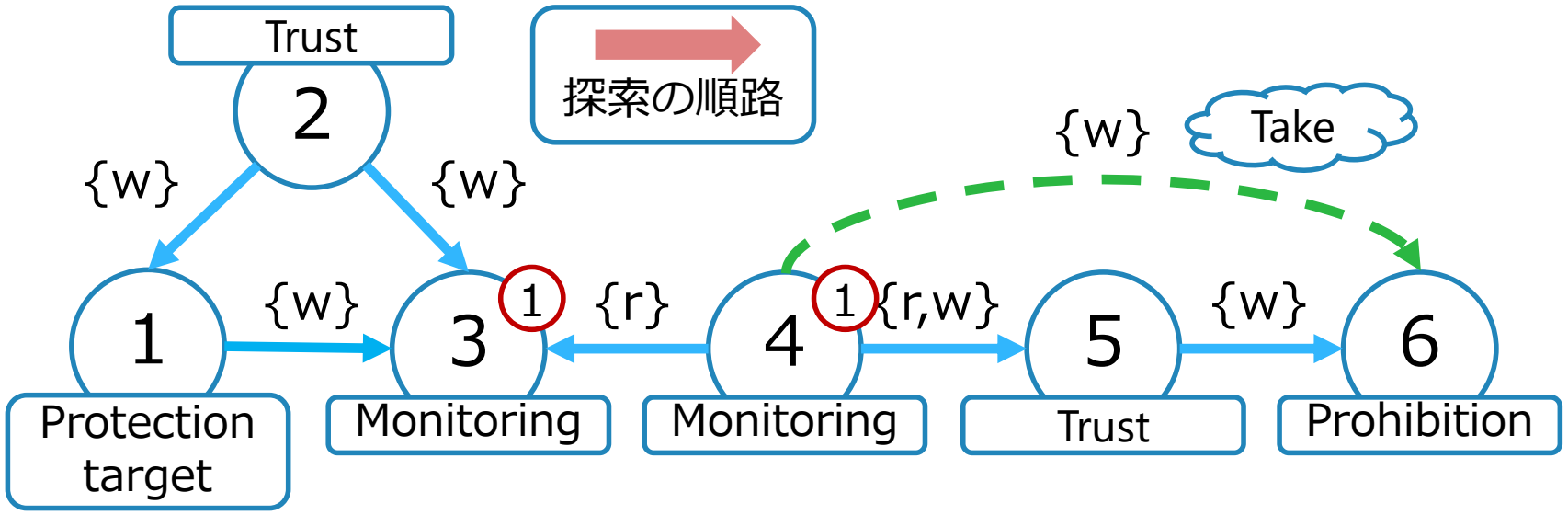
最終的に訪れるべきノードがなくなったため、Covert channelの発生していないことを示す

DRMのためのTake-Grant Model



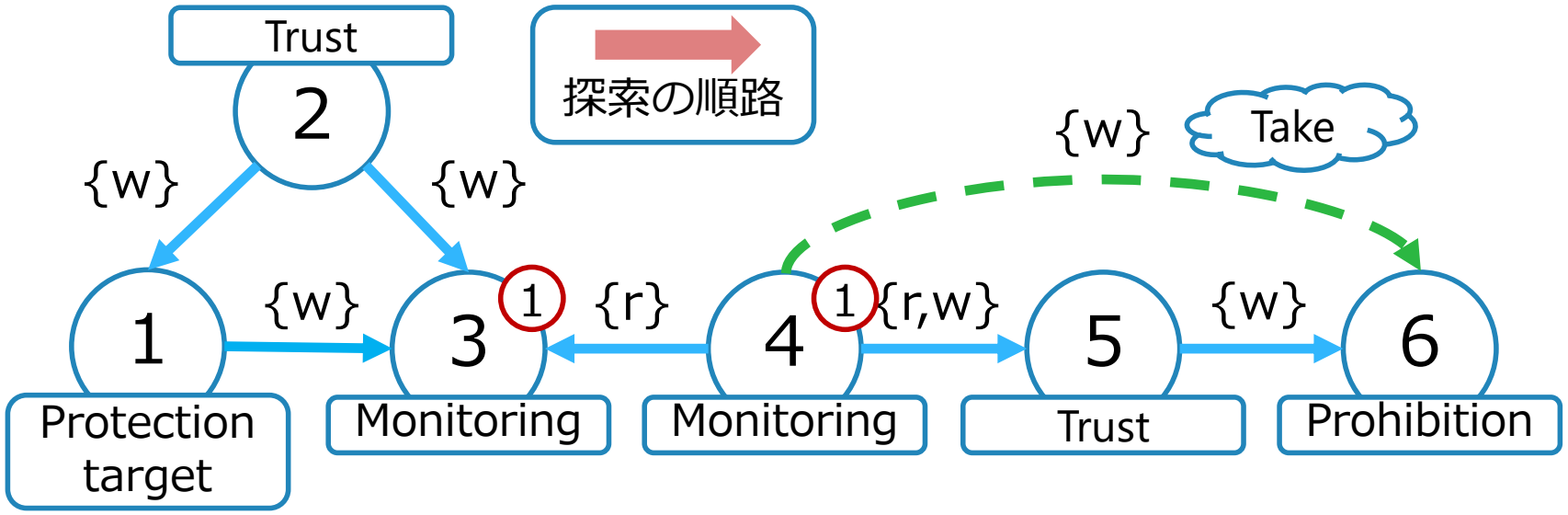
最後、探索経路となったすべてのノードに保護対象ノード1の保持を示すフラグを立てる

DRMのためのTake-Grant Model



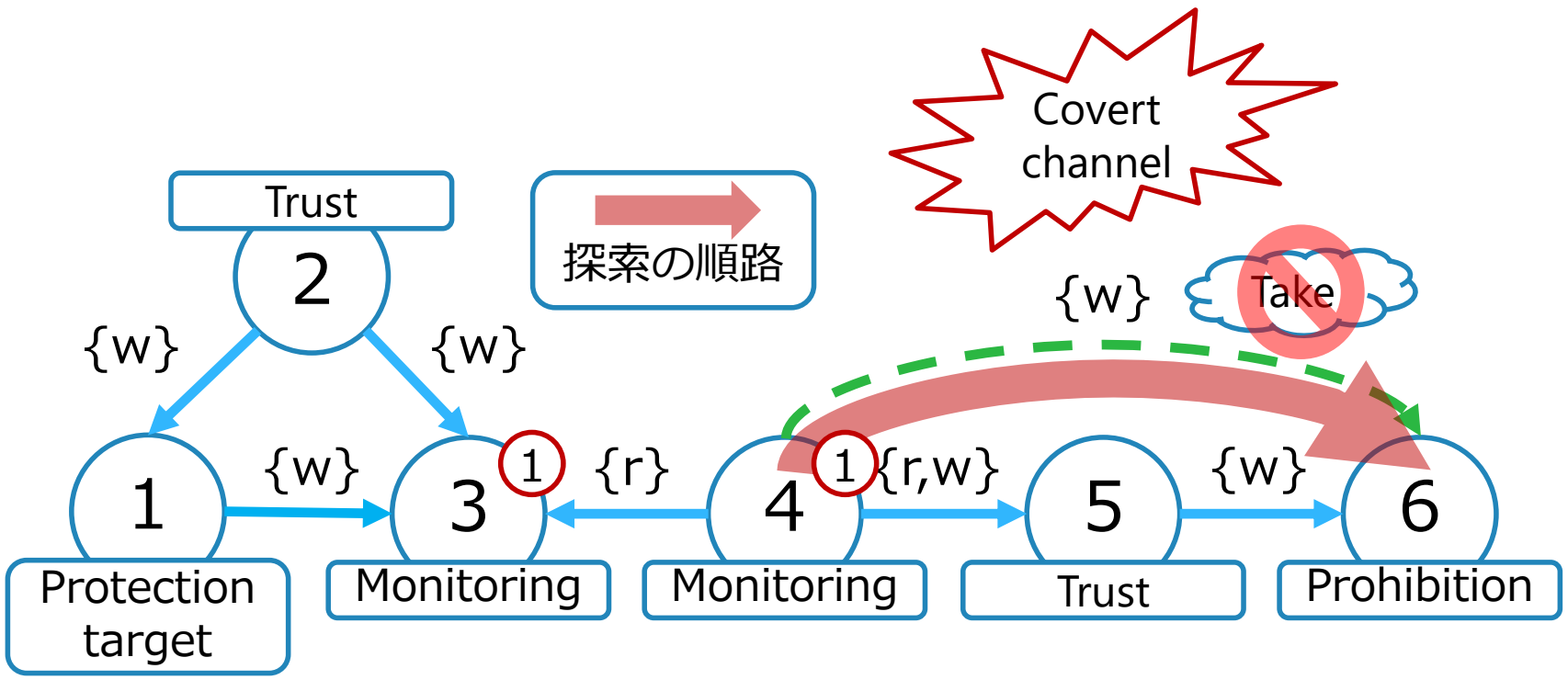
続いて、Takeによってノード4,6間に新たな辺 (ラベルWriteを持つ) を生成する場合を示す

DRMのためのTake-Grant Model



ノード4は保護対象ノード1の情報を保持を示すフラグを持つので起点となり、ノード１の Covert channel 発生を探索する

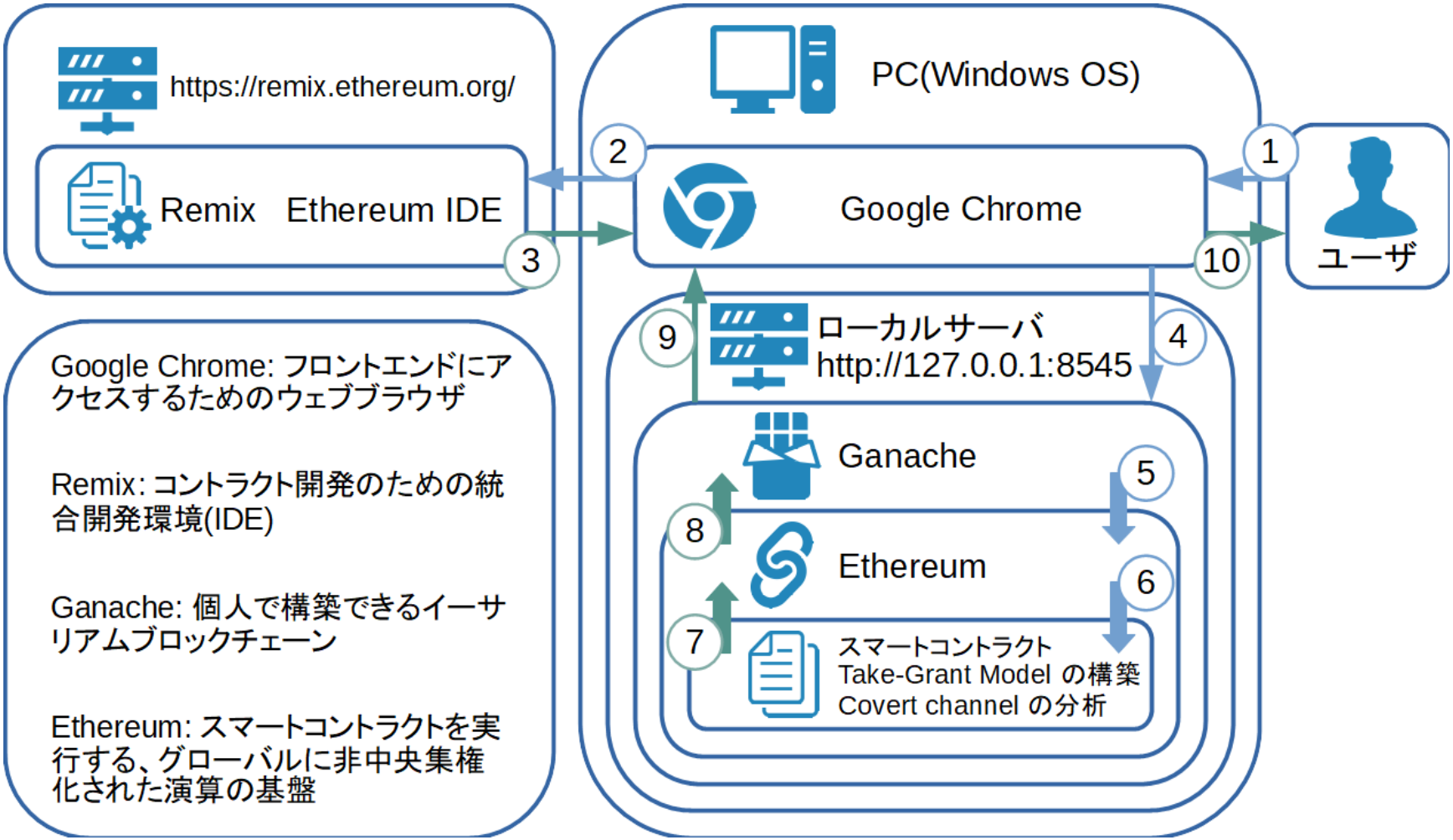
DRMのためのTake-Grant Model



禁止ノード6に辿り着いたことでCovert channel の発生を
検知、不正利用を防ぐためGrantの中断が行われる

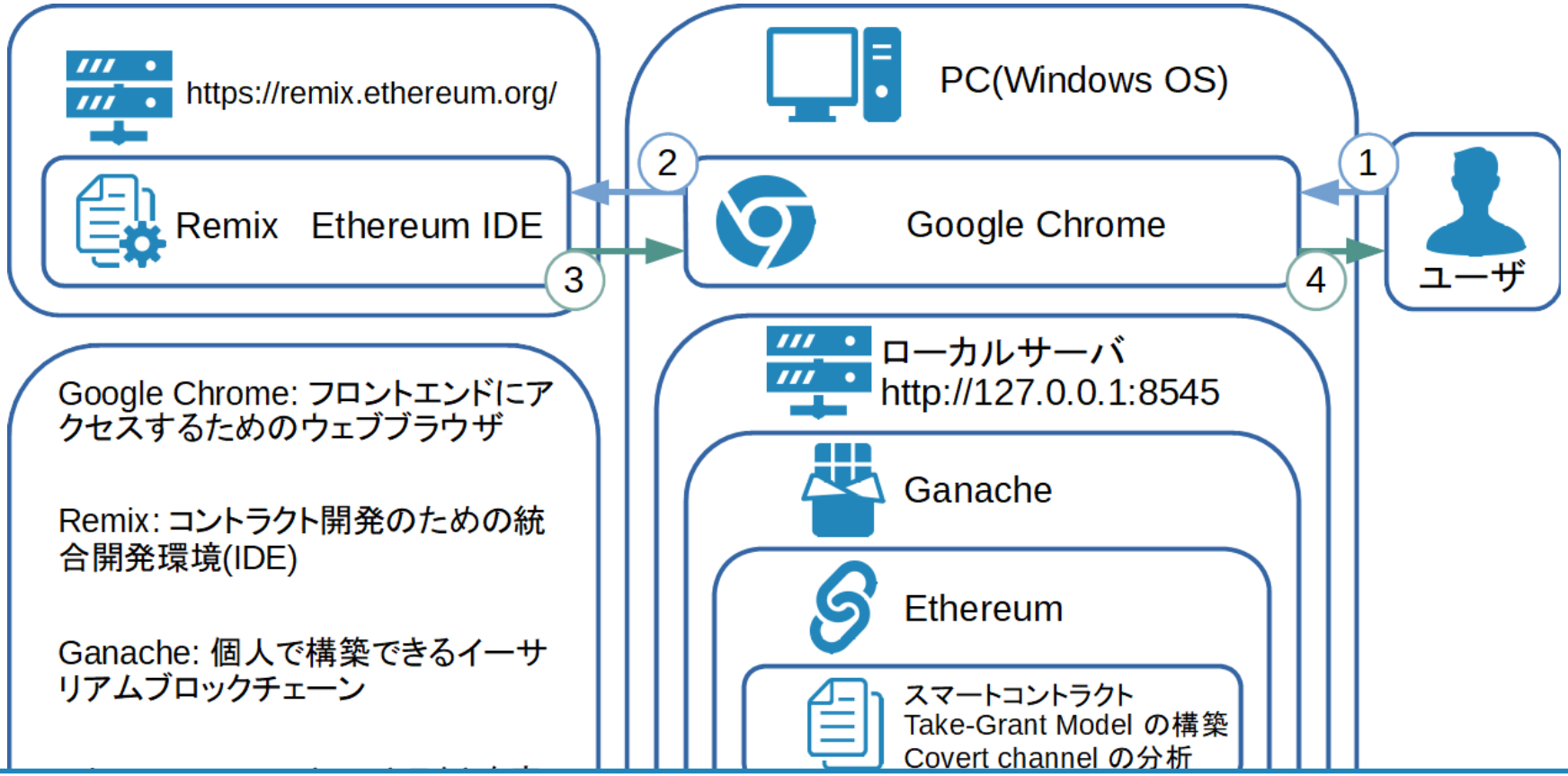
神奈川大学　松尾 浩貴

実装システム概要



神奈川大学　松尾 浩貴

実装システム概要

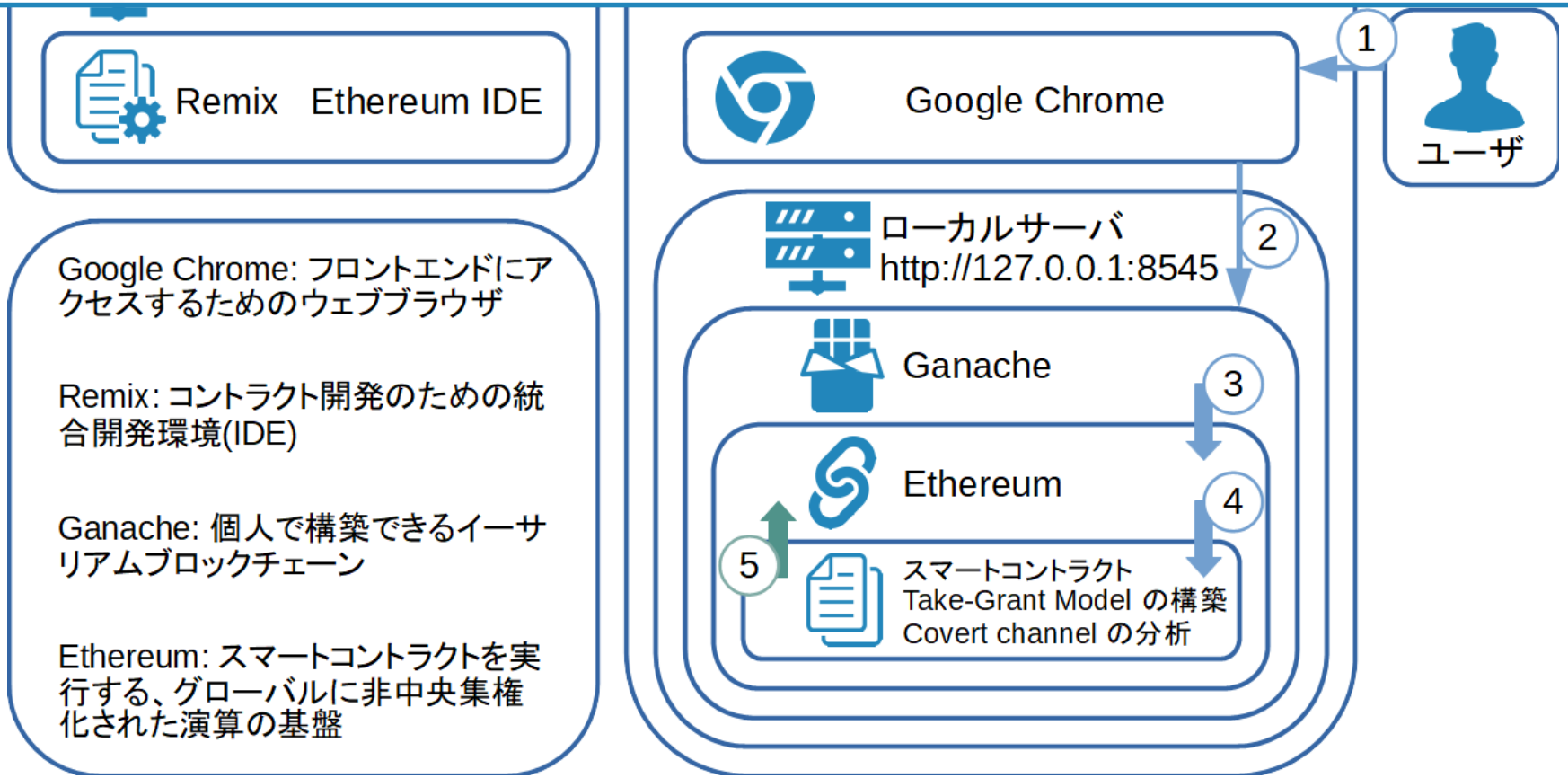


ユーザは基本操作を行うための必要情報をRemixに入力し
スマートコントラクト内の権限委譲を行う関数の実行を命令

神奈川大学　松尾 浩貴

実装システム概要

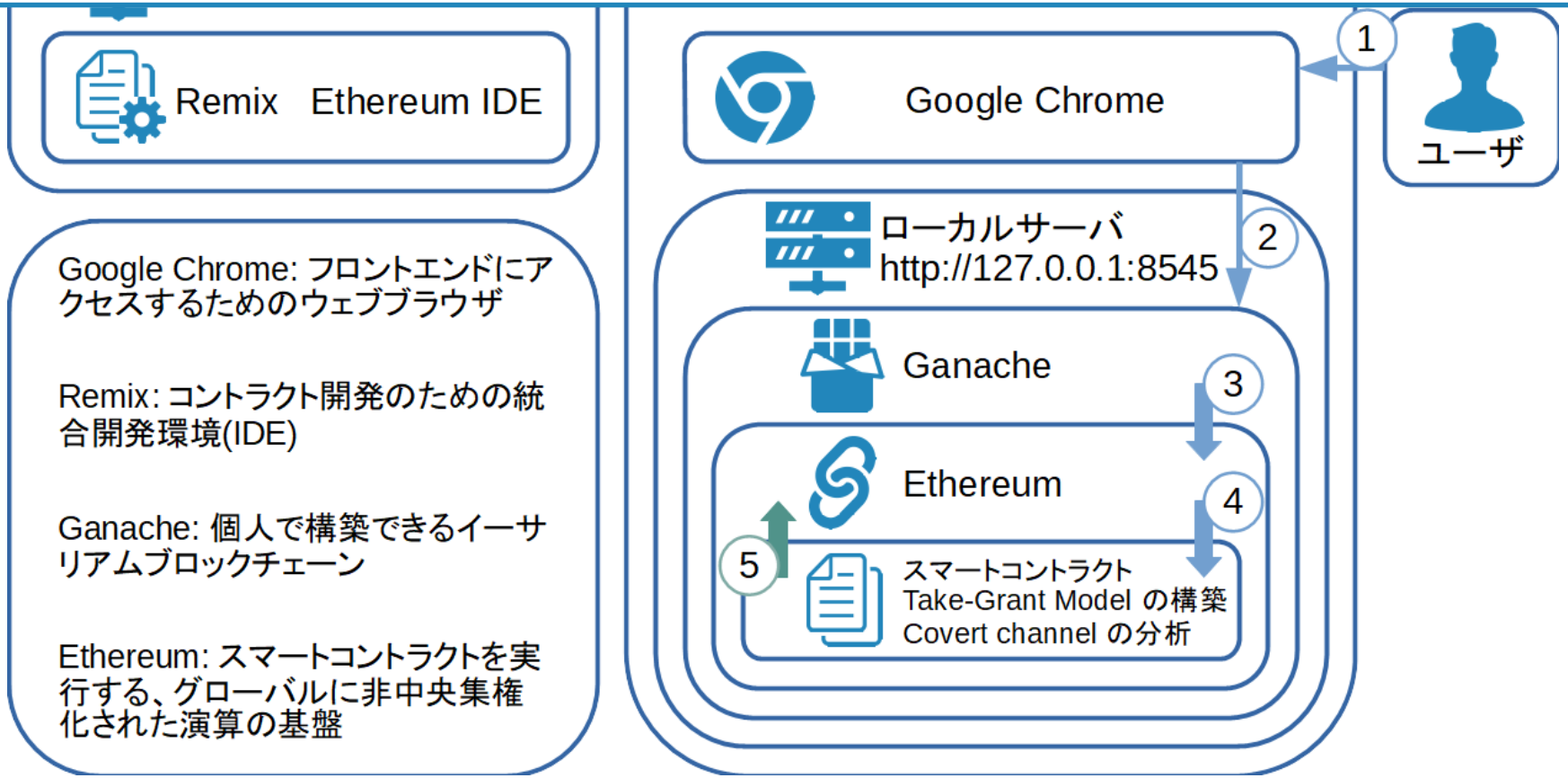
スマートコントラクト内の分析システムは、送られた情報をもとに権限委譲が履行できるかを分析を行う



神奈川大学　松尾 浩貴

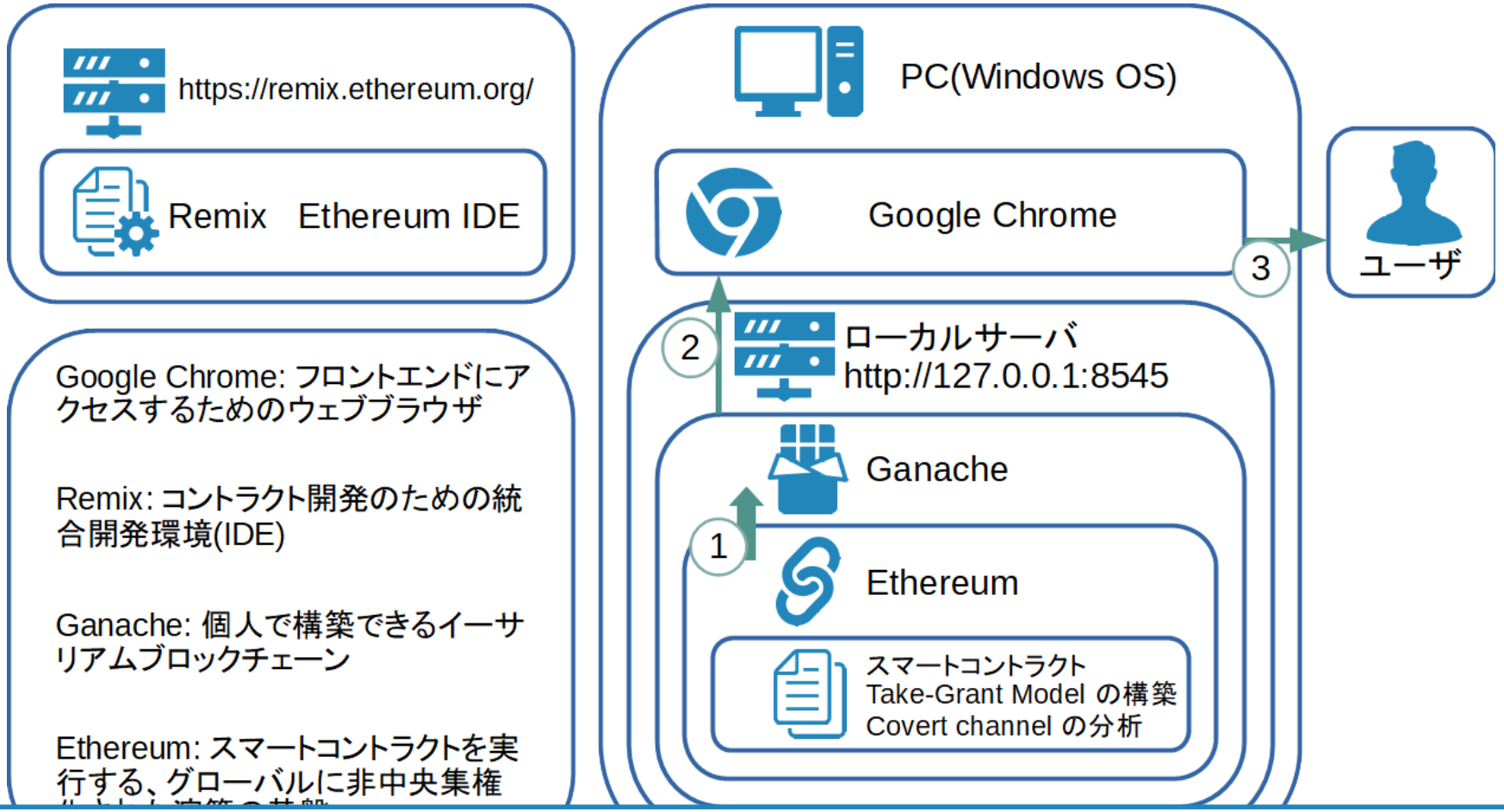
実装システム概要

分析の結果、権利委譲の履行条件を満たしていた場合は
取引結果をBCに記録、満たしていない場合は中断される



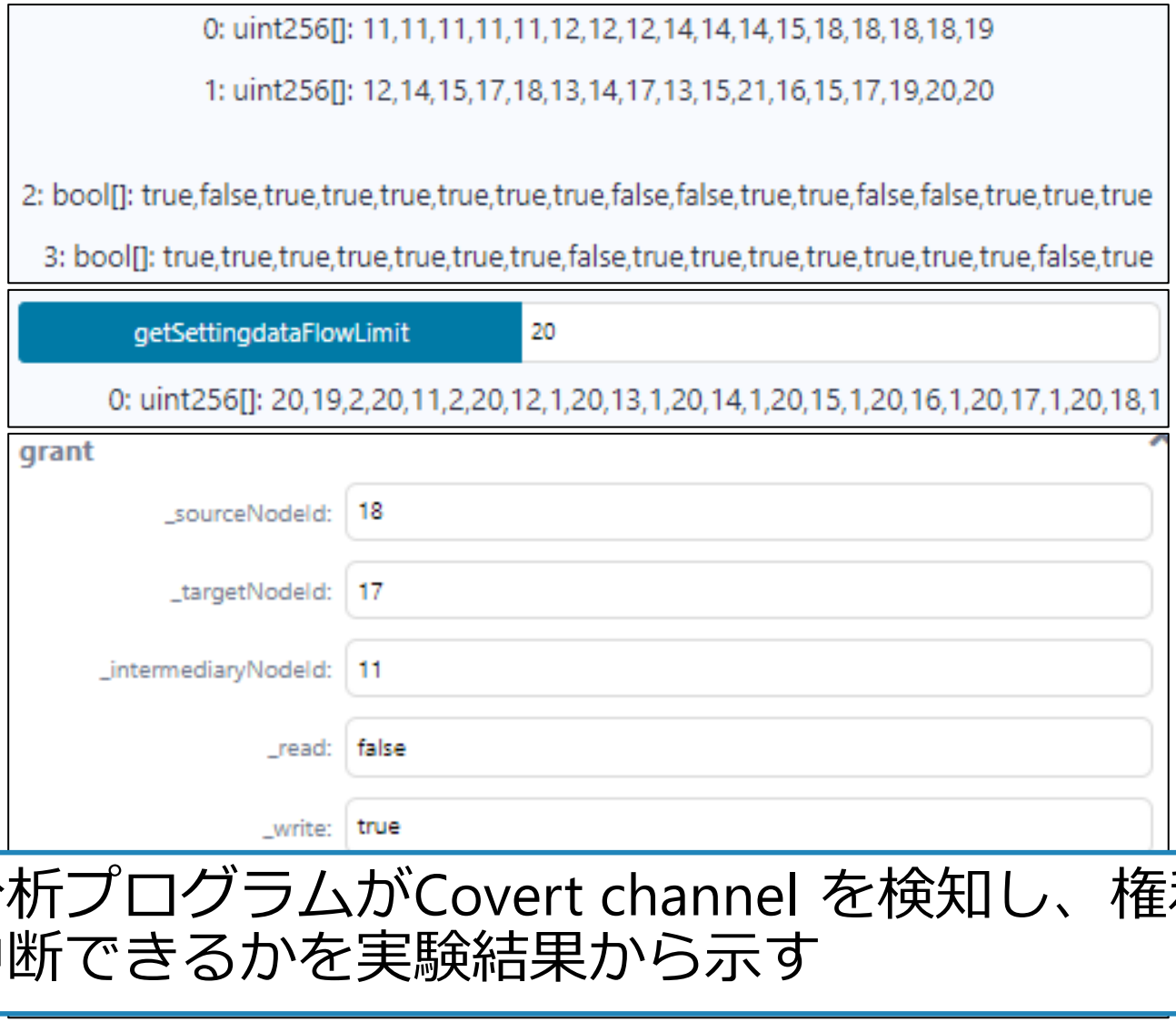
神奈川大学　松尾 浩貴

実装システム概要



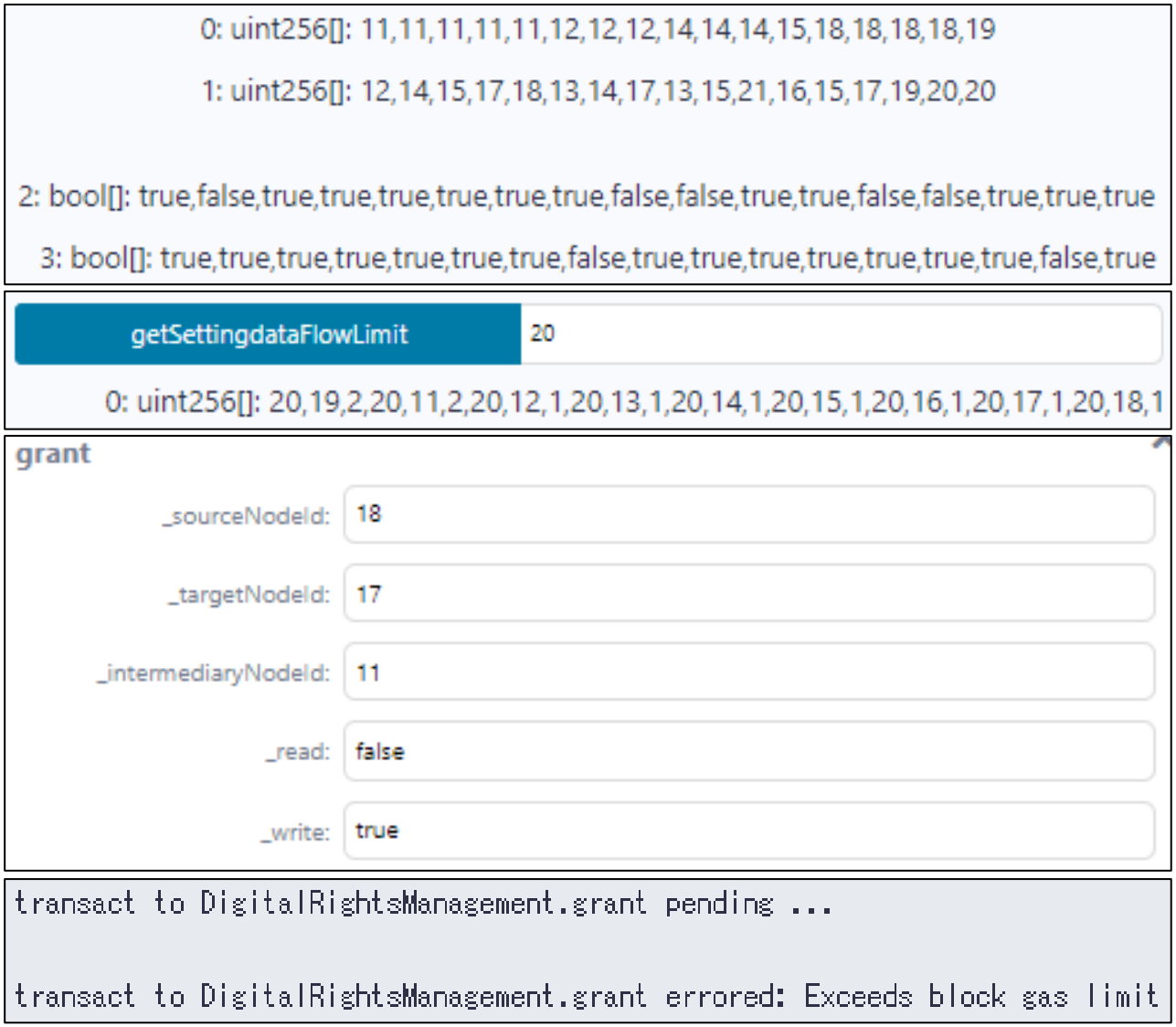
ユーザはRemixの更新にて、権限委譲の履行を確認する

実験



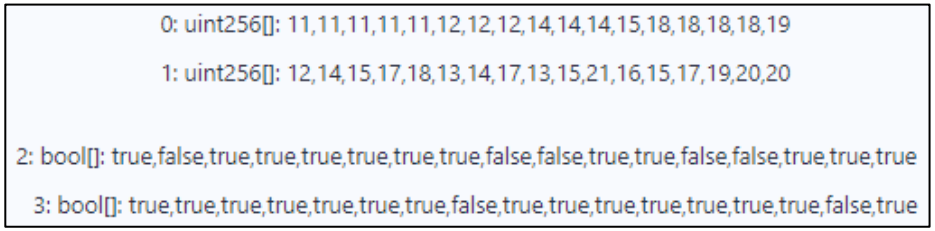
実際に分析プログラムがCovert channel を検知し、権利の委譲を中断できるかを実験結果から示す

実験

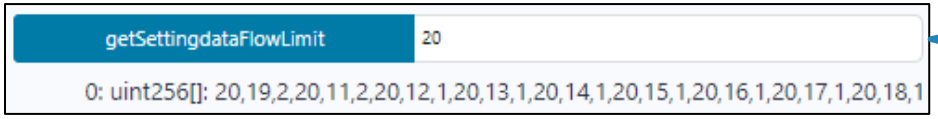


神奈川大学　松尾 浩貴

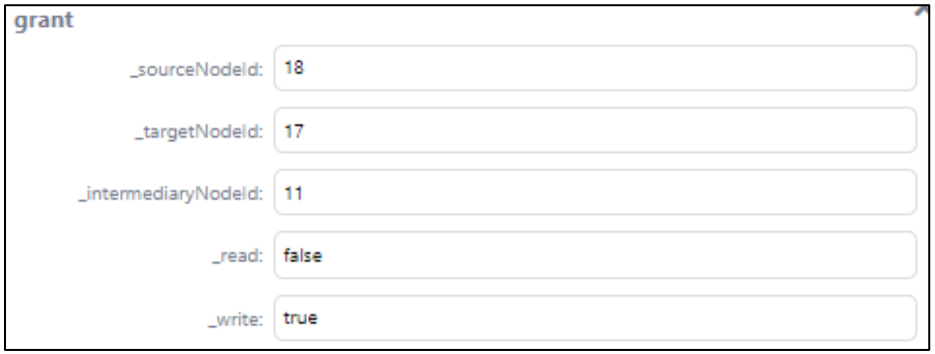
実験



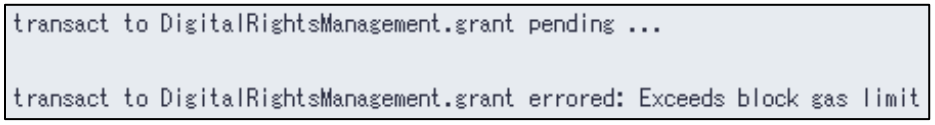
権利委譲の操作を行う前のグラフの状態



保護対象ノードが設ける他ノードに対する状態

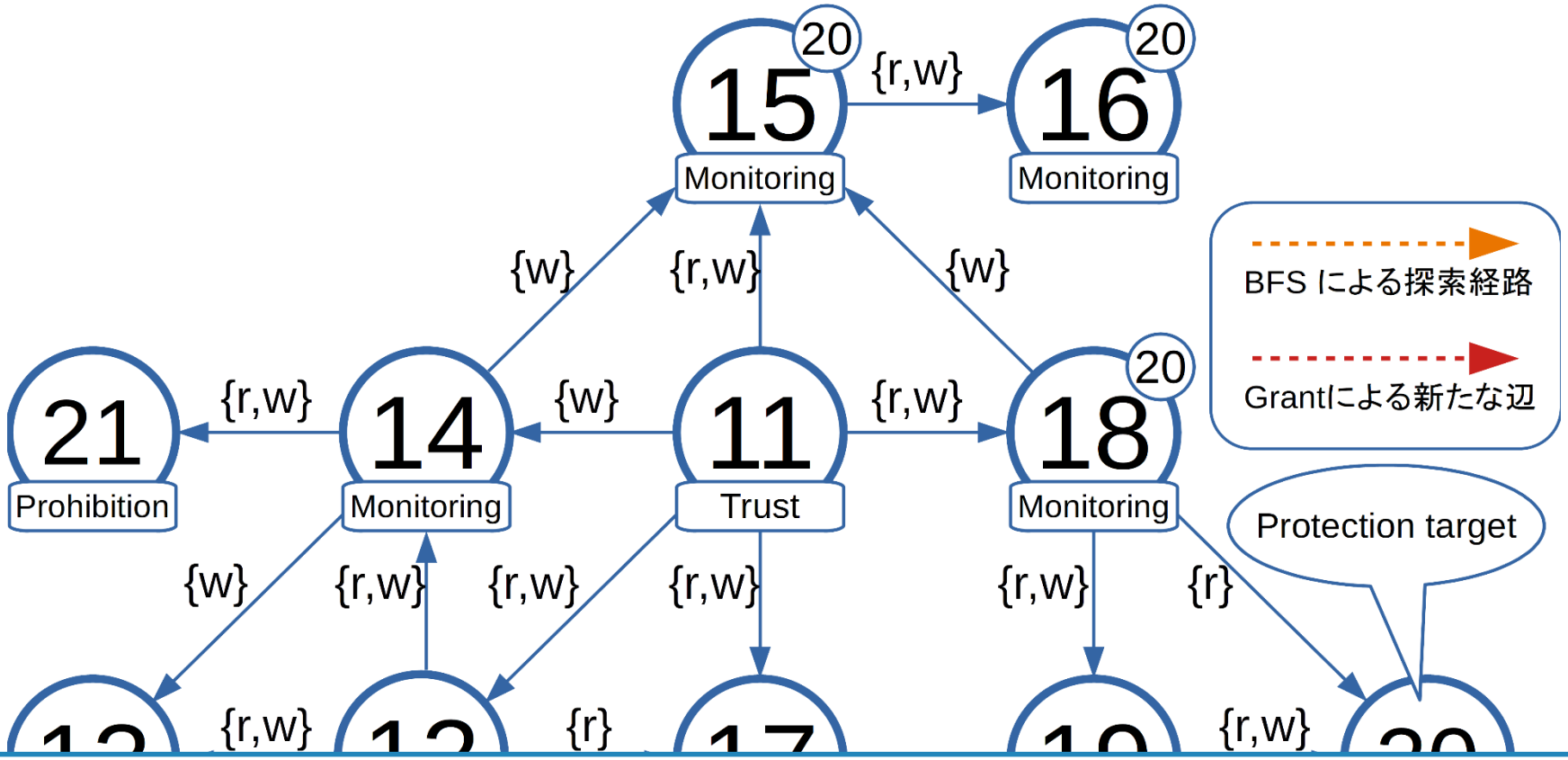


権利委譲の操作Grantの実行



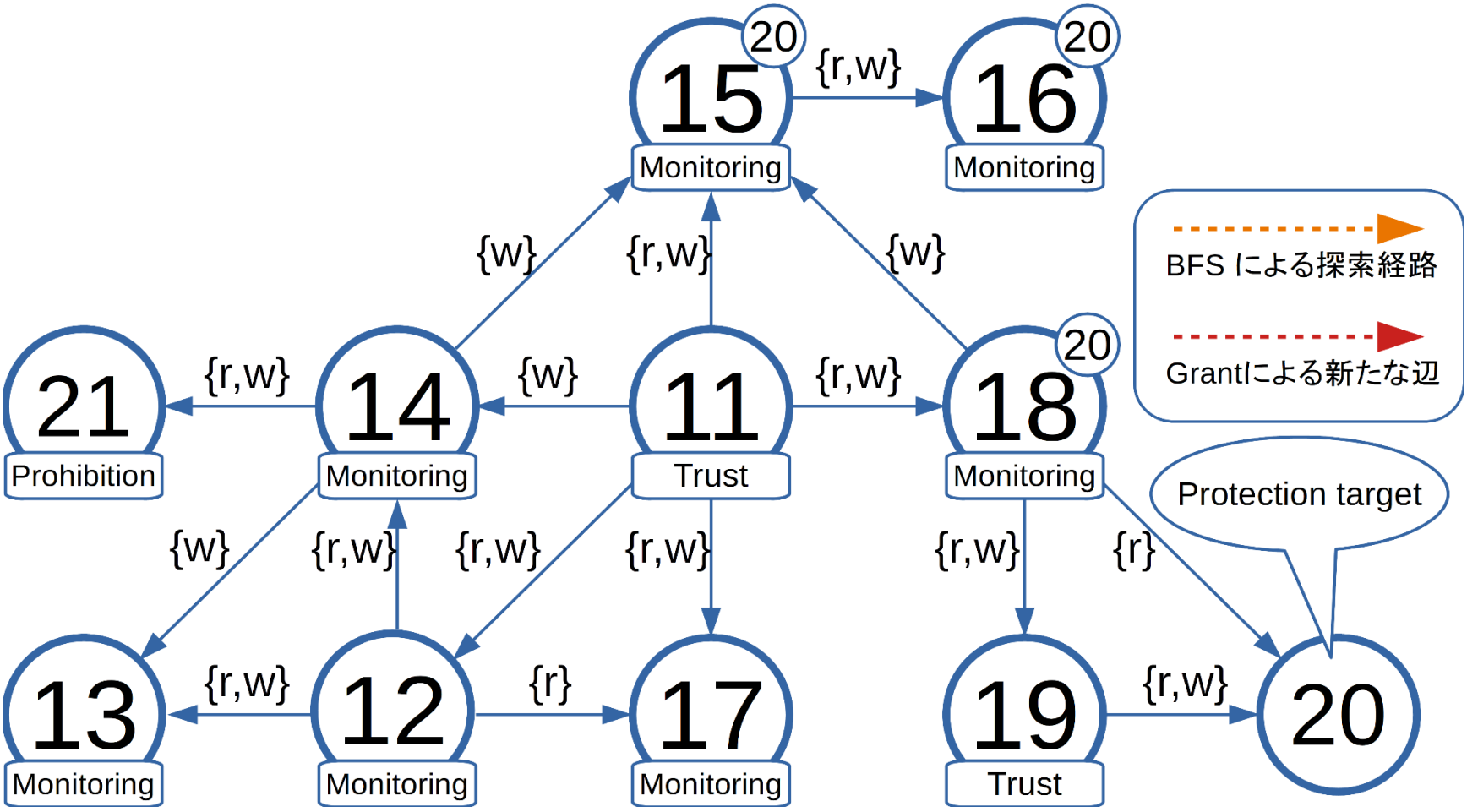
Grantがエラーで中断されたことを示すログ

実験

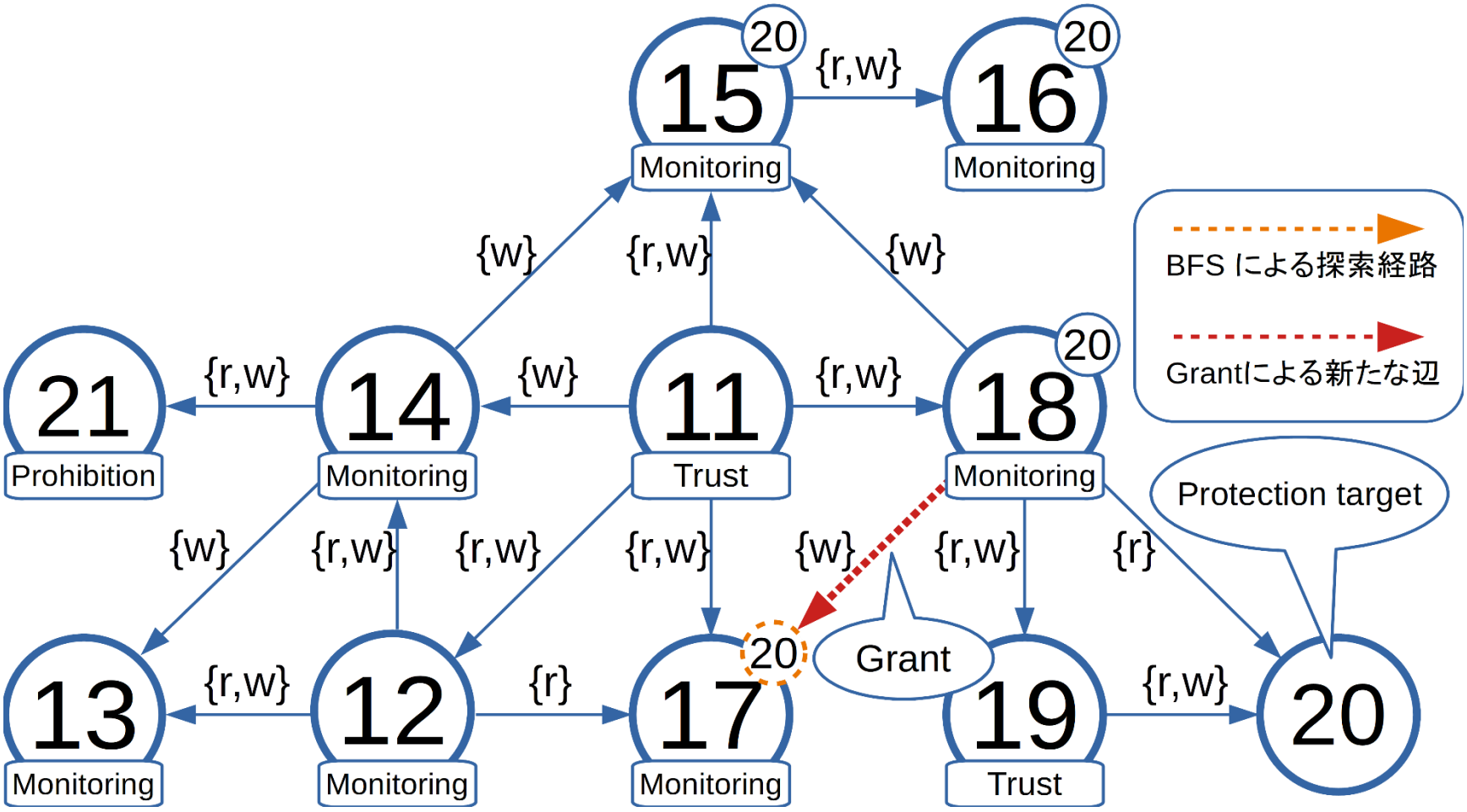


実験結果である文字の羅列だけではグラフの様子が分かりづらいので図にして表現する

実験

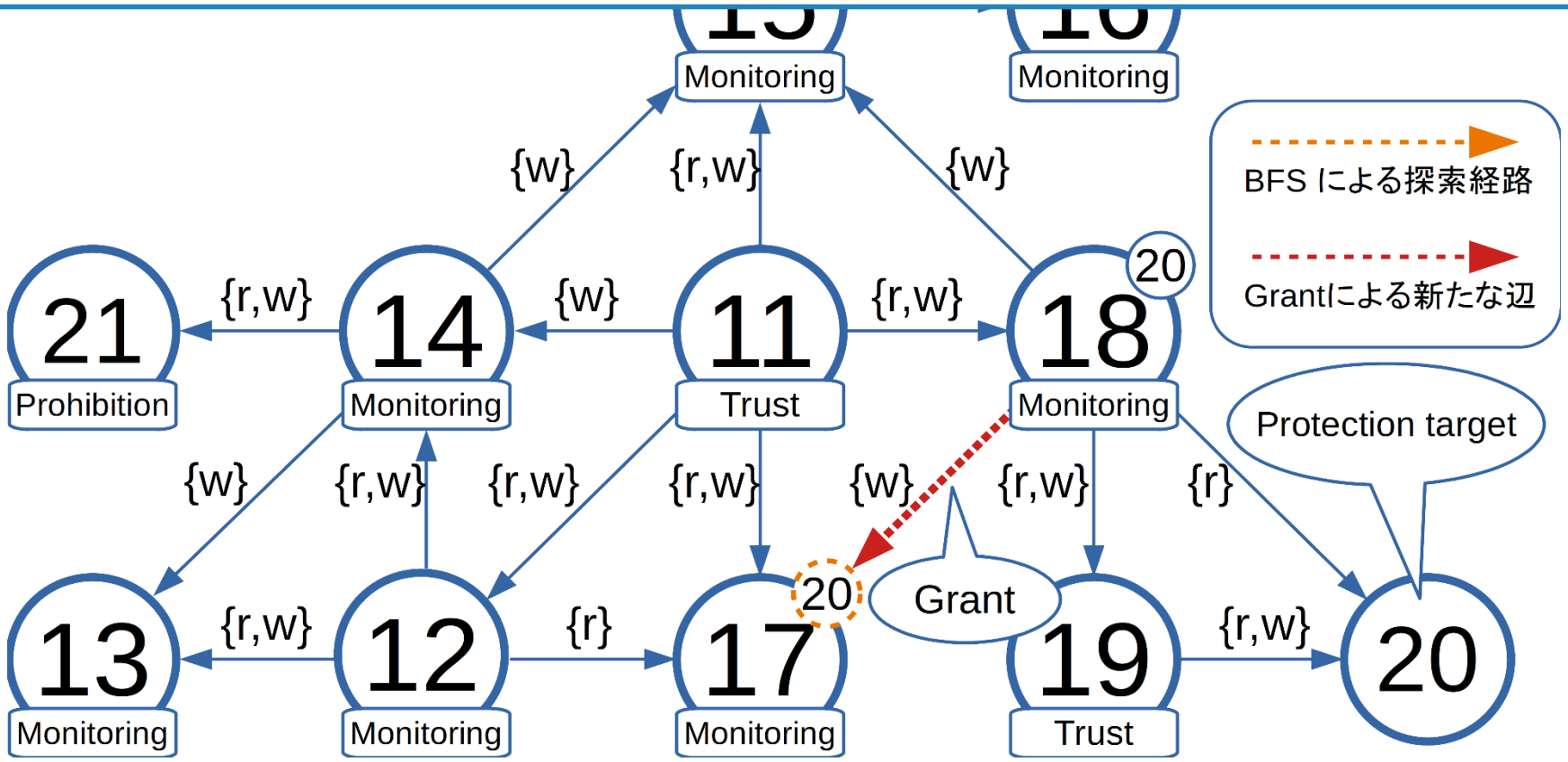

20
15
Monitoring
{r,w}
20
16
Monitoring
{w}
{r,w}
{w}
BFS による探索経路
Grantによる新たな辺
21
Prohibition
{r,w}
14
Monitoring
{w}
11
Trust
{r,w}
20
18
Monitoring
Protection target
{w}
{r,w}
{r,w}
{r,w}
{r,w}
{r}
13
Monitoring
{r,w}
12
Monitoring
{r}
17
Monitoring
19
Trust
{r,w}
20

実験


20
15
Monitoring
{r,w}
20
16
Monitoring
{w}
{r,w}
{w}
BFS による探索経路
Grantによる新たな辺
21
Prohibition
{r,w}
14
Monitoring
{w}
11
Trust
{r,w}
20
18
Monitoring
Protection target
{w}
{r,w}
{r,w}
{r,w}
{w}
{r,w}
{r}
20
Grant
13
Monitoring
{r,w}
12
Monitoring
{r}
17
Monitoring
19
Trust
{r,w}
20

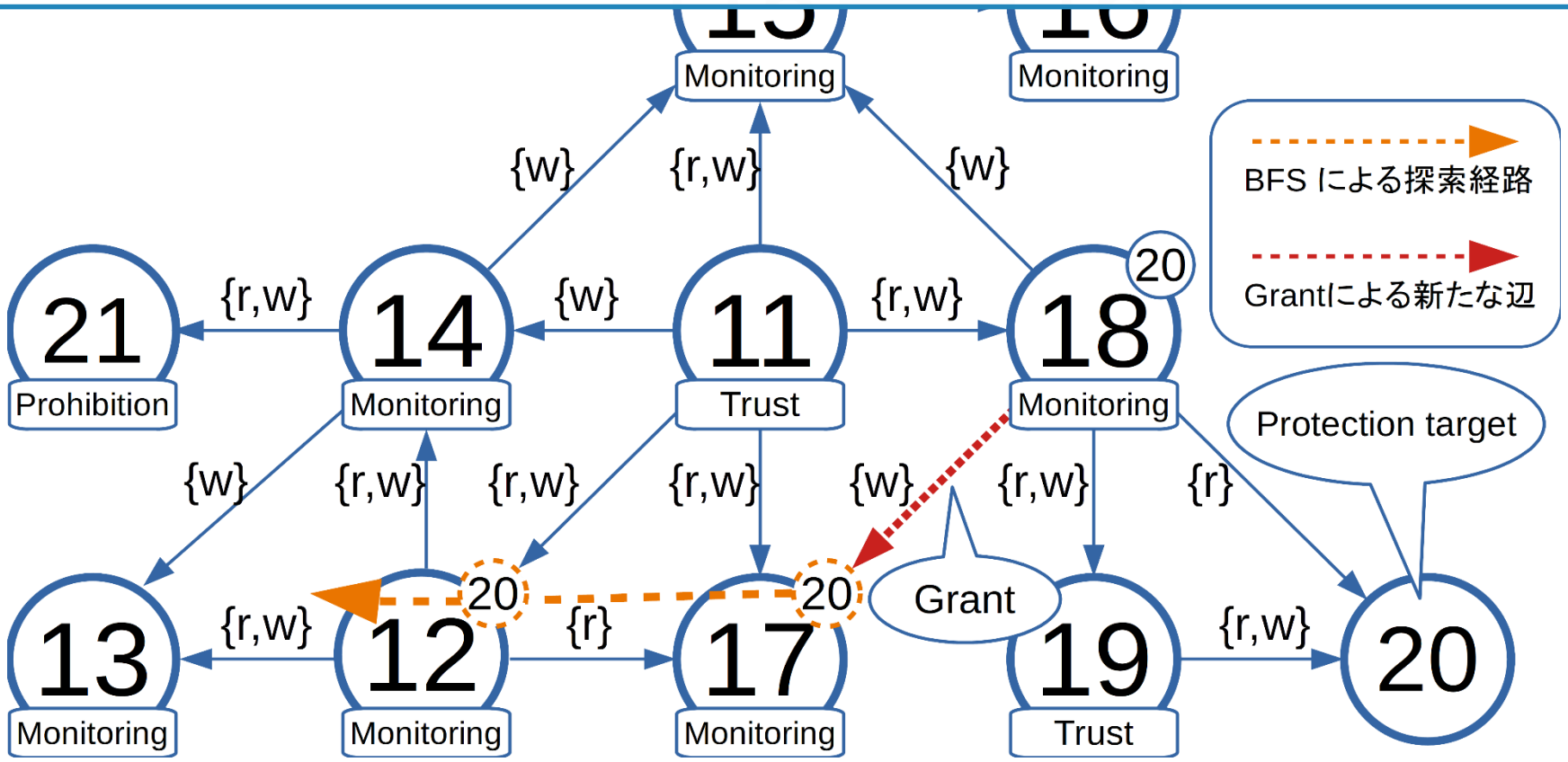
実験

ノード18は保護対象ノード20を持っていることにより、
新たな辺発生時、Covert channel 起こる方向へBFS



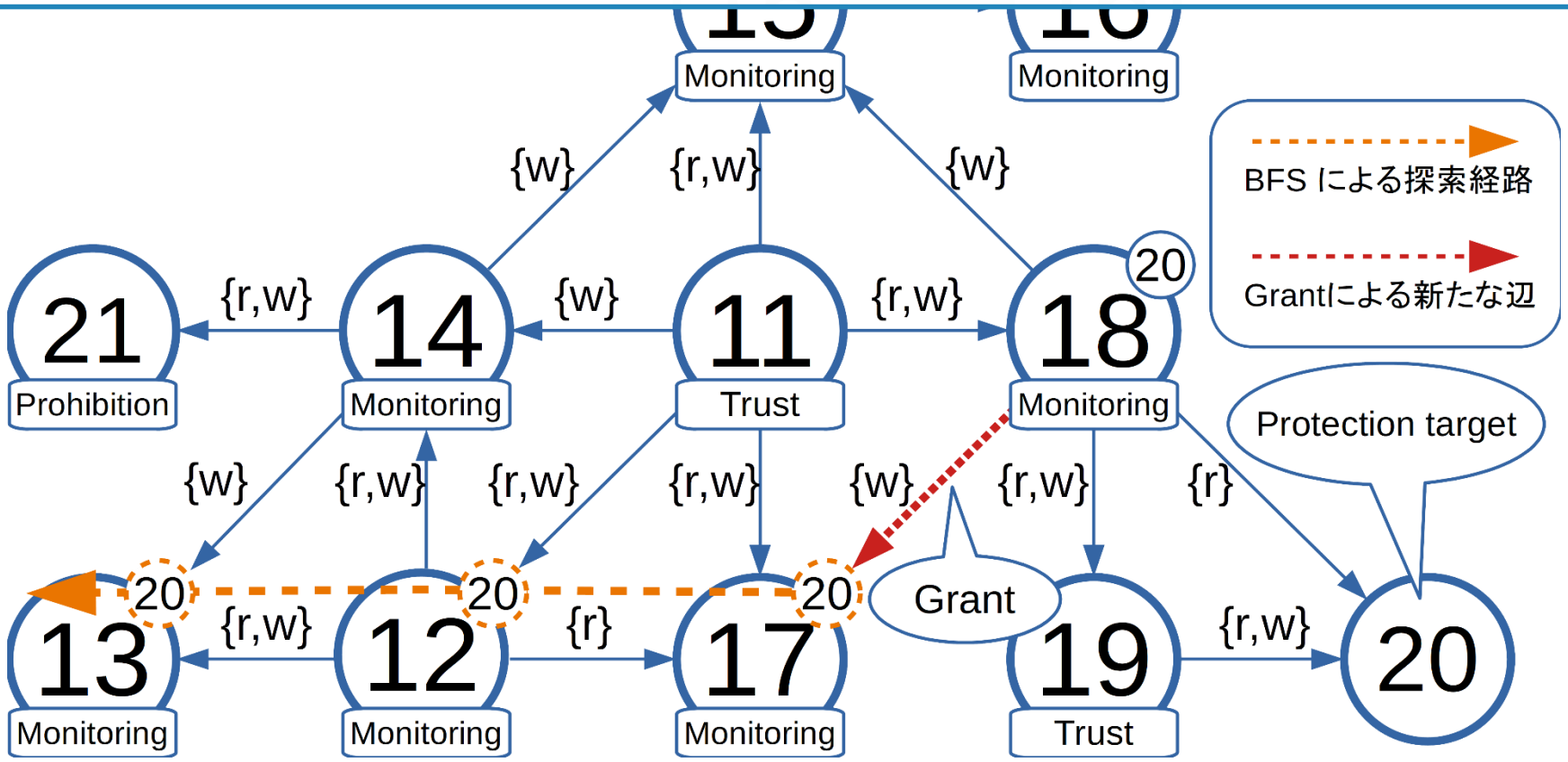
実験

ノード11は保護対象ノード20の課す制限が信頼ノードである為、それ以降に起こるであろうCovert channel は無視される

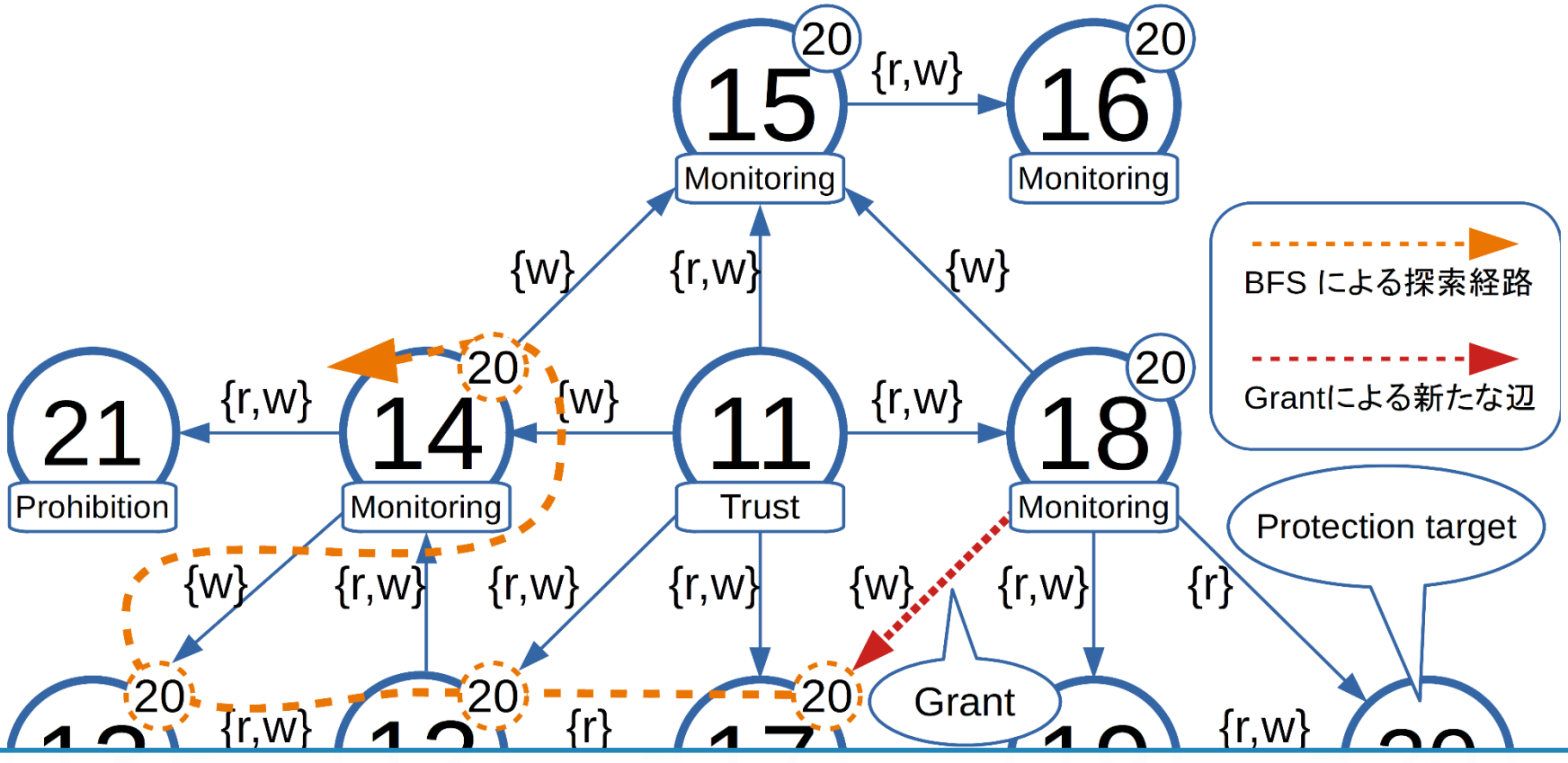


実験

ノード12,13は保護対象ノード20の課す制限が監視ノードのため、情報の所持は許されるが、BFSの探索経路になる



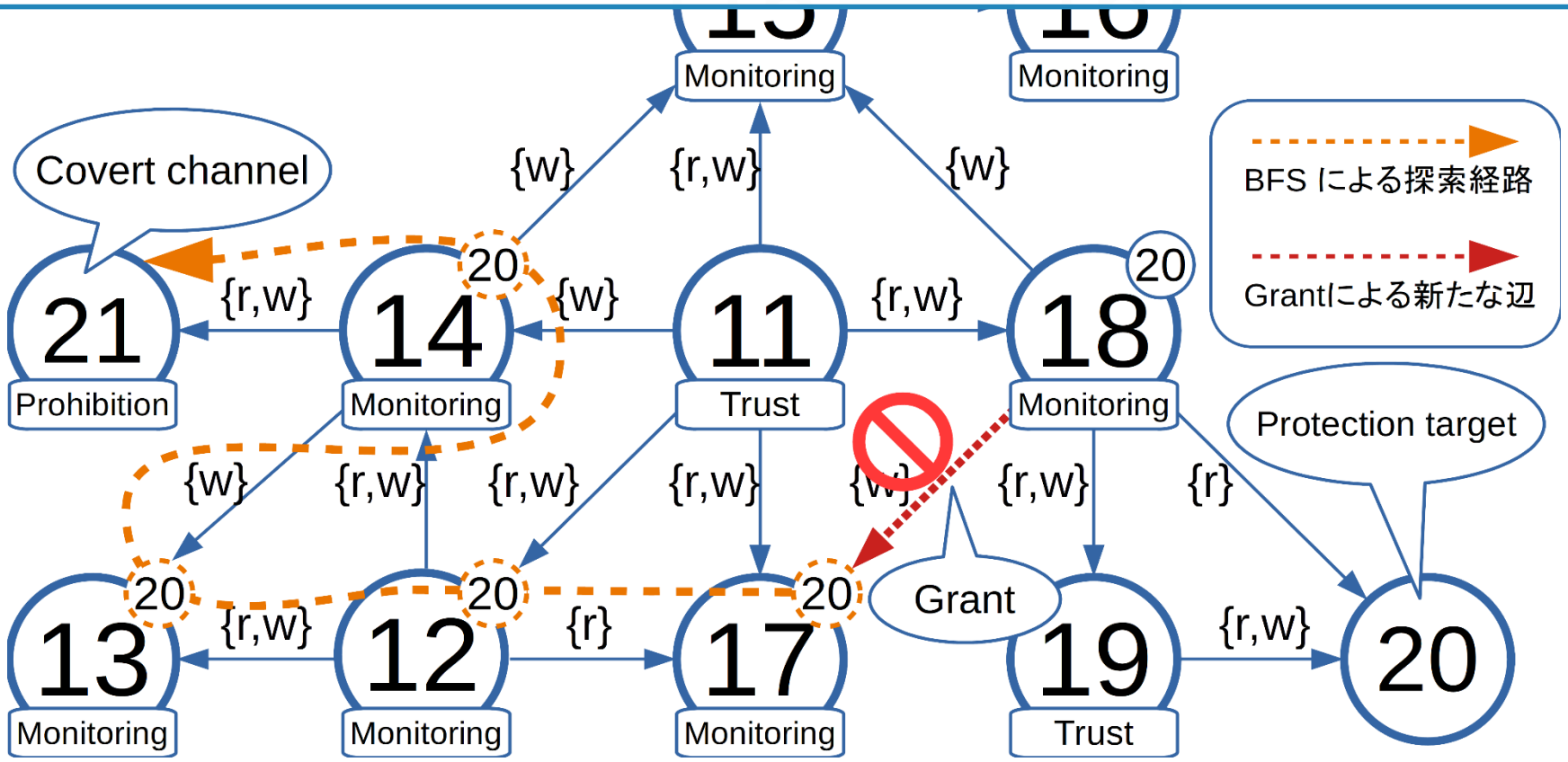
実験



ノード15は保護対象ノード20の情報をすでに所持している
ことが分かるので探索対象にならない

実験

ノード21は保護対象ノード20の課す制限が禁止ノードである為、Covert channel 検知し、権利委譲は中断される



結論・今後の課題

結論

- Covert channelの発生をデジタルコンテンツの不正利用にモデル化し、不正利用を防ぐことによって新たなDRMを提案した
- 従来の DRM よりもデジタルコンテンツの著作者の権利保護と安全なコンテンツの流通を確立できた

今後の課題

- デジタルコンテンツおよび利用者とTake – Grant Modelにおける有向グラフのノードとの紐付け
- 基本操作を行うごとにBFSを行っているので非常に計算コスト（運用に掛かる実費）が大きい。よって、さらに効率のよいスマートコントラクトを構築するか、そもそもTake – Grant Modelの解析をスマートコントラクト内ではなく、外部に設置することも検討する必要がある

質疑応答

- ご清聴ありがとうございました

神奈川大学　松尾 浩貴

DRM(デジタル著作権管理)

- 「Digital Rights Management」の略
- DRMとは、デジタルデータの著作権を保護するため、利用や複製の制限をかける技術の総称
- 複製制限技術、電子透かしや暗号化 等が存在する

ブロックチェーンとは

- ブロックチェーンとは、分散型台帳技術または分散型ネットワークである

- 同じデータを分散させて管理する

- データの塊であるブロックはハッシュ関数によって暗号化され、ブロック同士が鎖のように繋がっている

- ブロックチェーンは優れた改ざん耐性を持つ

スマートプロパティ

- スマートプロパティとは、デジタルコンテンツや不動産などの資産(プロパティ)をネットワーク上で流通できるようにしたもの

- 資産の情報を含むコインに見立てたデータを生成

- そのコインを用いて資産情報の保護、所有者を表し流通をさせる

神奈川大学　松尾 浩貴

Ethereum(イーサリアム)

- Ethereum(イーサリアム)とは分散型アプリケーションやスマートコントラクトを構築するためのプラットフォーム
- 非中央集権であり、イーサリアムネットワークと呼ばれるP2Pのネットワークをもつ
- ネットワーク参加者はブロックチェーンに任意のスマートコントラクトを記述し、実行、履行履歴を記録することができる
- スマートコントラクトはチューリング完全なプログラム言語で記述される

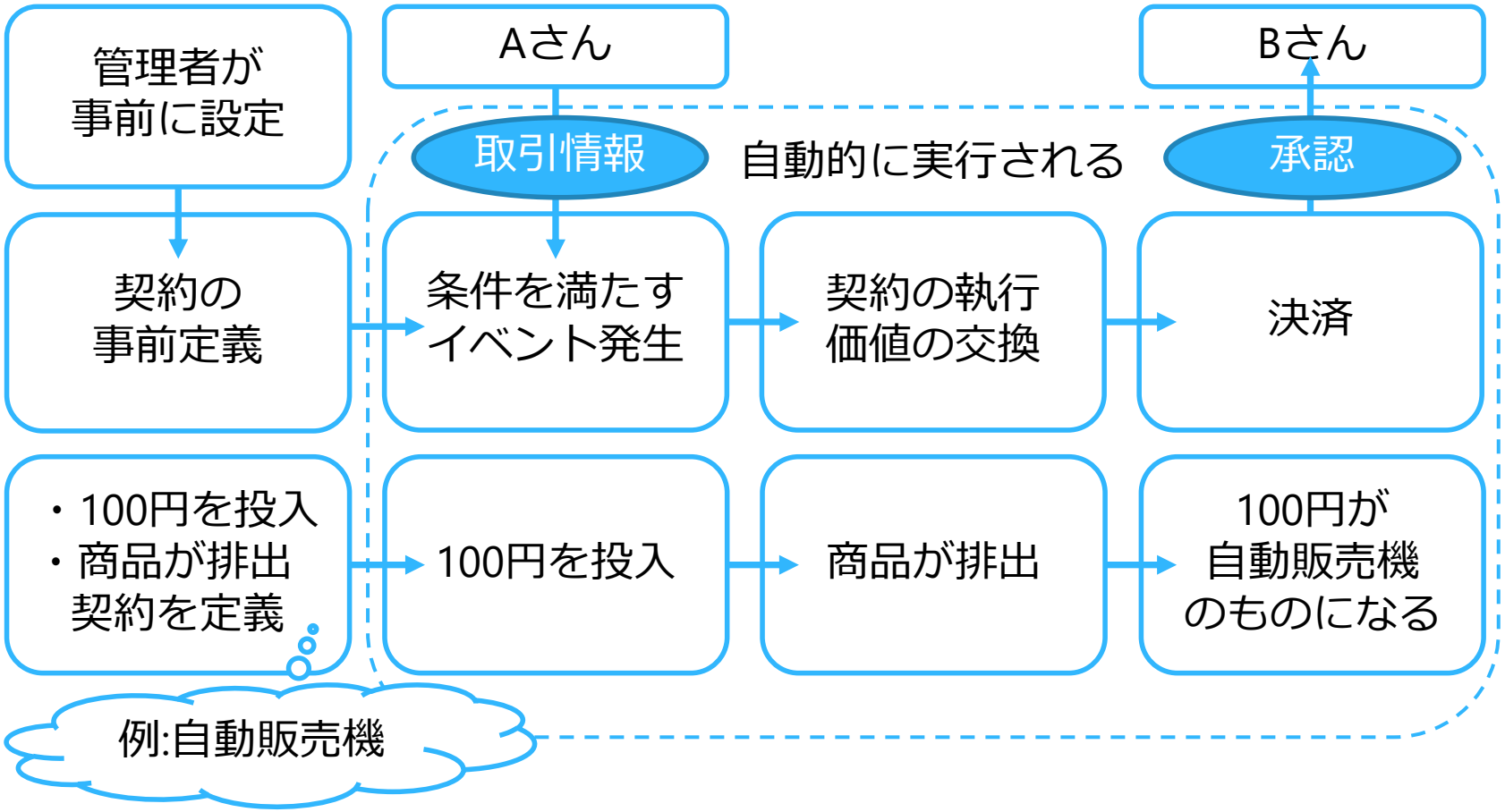
スマートコントラクト

- 「スマート」＝「賢い」、「コントラクト」＝「契約」

- スマートコントラクトとは、
  契約を自動化することであり、契約の条件確認や
  履行までを自動的に実行することができる

- 取引を自動化できるため不正の防止できる
- 契約にかかるコストを削減
- 仲介者が不要
- 非中央集権のサービス

スマートコントラクト



定義された契約条件をブロックチェーンで管理するため
改ざんの心配なく自動的に取引を行うことができる

神奈川大学　松尾 浩貴

Meta Mask と Web3.js

- Meta Mask とは、Google Chrome のプラグインのひとつ
- ウェブサイトの Javascript に対して、EthereumのWeb3 API を提供し、 ブロックチェーンのデータをブラウザから読むことができる
- また、アカウントを管理し、ブロックチェーントランザクションに署名するためのユーザーインターフェイスを提供する
- Web3.jsとは、HTTPやIPCを利用し、Ethereumネットワーク上のノードとローカルまたはリモートで通信を可能にする JavaScript API

Ganache

- Ganache は、Ethereum のローカル開発環境を提供するアプリである
- Ethereumのインストールなどが不要で、Ganache単体でEthereumネットワークの起動ができる
- 初期設定せずともテスト用アカウントを用意
- ブロック生成速度を簡単に変更できる
(マイニングコントロール)
- アカウント情報をすばやく視覚的に確認できる
(アドレス、秘密鍵、残高 等)

Geth

- Geth(Go Ethereum)
- Ethereumで公式に提供されているEthereumのための
  クライアントソフト
- Goというプログラミング言語で実装
  ※GoはGoogleによって開発されたプログラミング言語
- Gethはコマンドプロンプト上で、
  スマートコントラクトの生成や実行、etherの送金、
  アカウント作成、マイニング etc.
  Ethereumで必要なことがほぼできる

神奈川大学　松尾 浩貴

Take-Grant Model

- Take-Grant Model は次の要素からなる有限の有向グラフによってモデル化される

- 主体の集合と対象の集合は互いに素であるとする
- ラベル r は読み取り、w は書き込みの権限とする

- 頂点
  主体または対象。
- ラベル付きの有向辺
  主体および対象が持つ許可
  ラベルは権限を表し{r}, {w}, {r, w}のいずれかである

Take-Grant Model

Take

- 一定の条件のもとで、主体が他の主体または対象の持つ権限を獲得する

グラフ G に相異なる頂点 x, y, z が存在して以下の条件

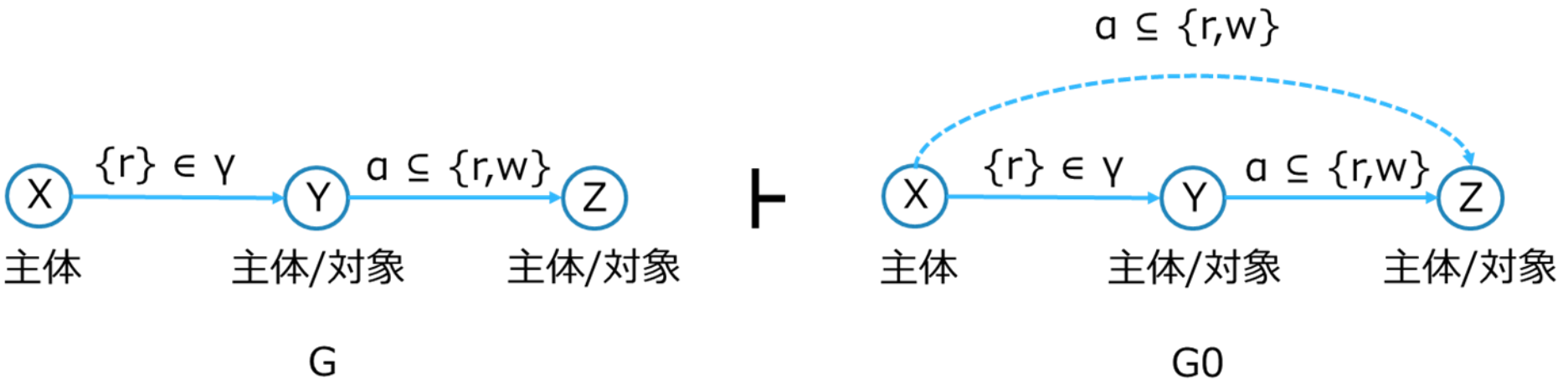
- x は主体、y と z は主体または対象である
- x から y へラベル γ の辺が存在し、r ∈ γ である
- y から z へラベル α⊆{r, w}の辺が存在する

これらを満たすとき、G を次のように書換えたグラフ G0 を生成する

- x から z へラベル α の辺を追加する

Take-Grant Model

Take



Take-Grant Model

Grant

- 一定の条件のもとで、主体が他の主体または対象に権限を委譲する

グラフ G に相異なる頂点 x, y, z が存在して以下の条件

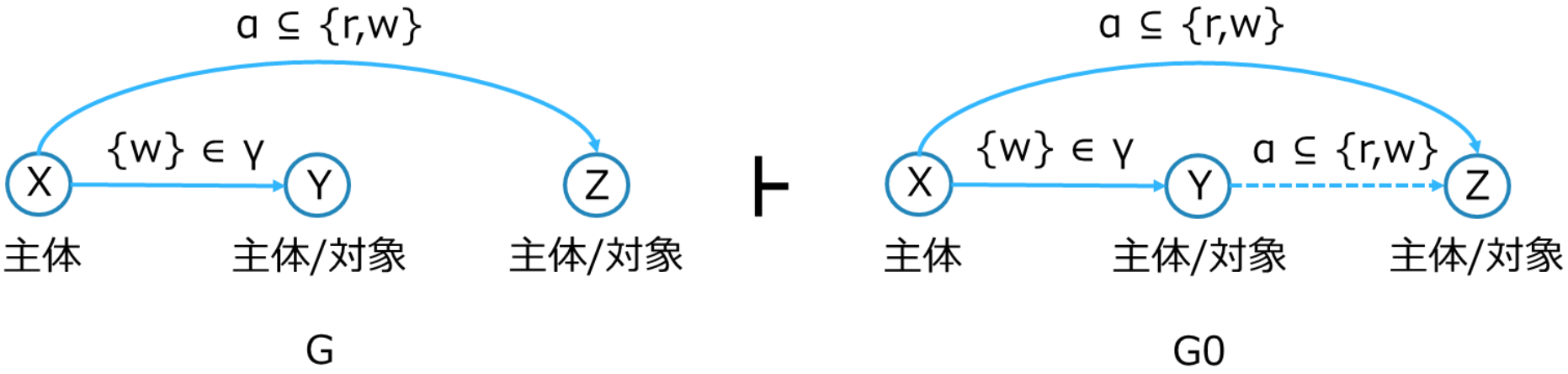
- x は主体、y と z は主体または対象である
- x から y へラベル γ の辺が存在し、w ∈ γである
- x から z へラベル α⊆{r, w}の辺が存在する

これらを満たすとき、G を次のように書換えたグラフ G0 を生成する

- y から z へラベル α の辺を追加する

Take-Grant Model

Grant



Take-Grant Model

Create

- 新たな主体を作成する

グラフ G に頂点 x が存在して以下の条件

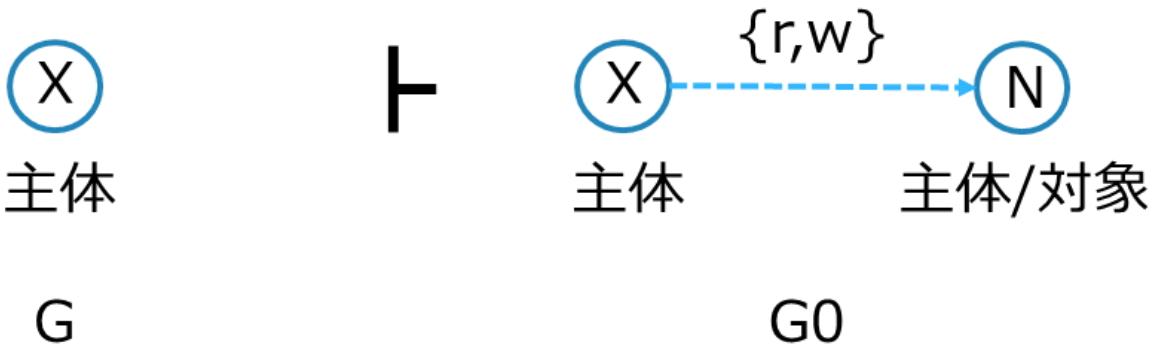
- x が主体である

これを満たすとき、G を次のように書換えたグラフ G0 を生成する

- 新しい主体の頂点 N を追加する x から N へラベル {r, w}の辺を追加する

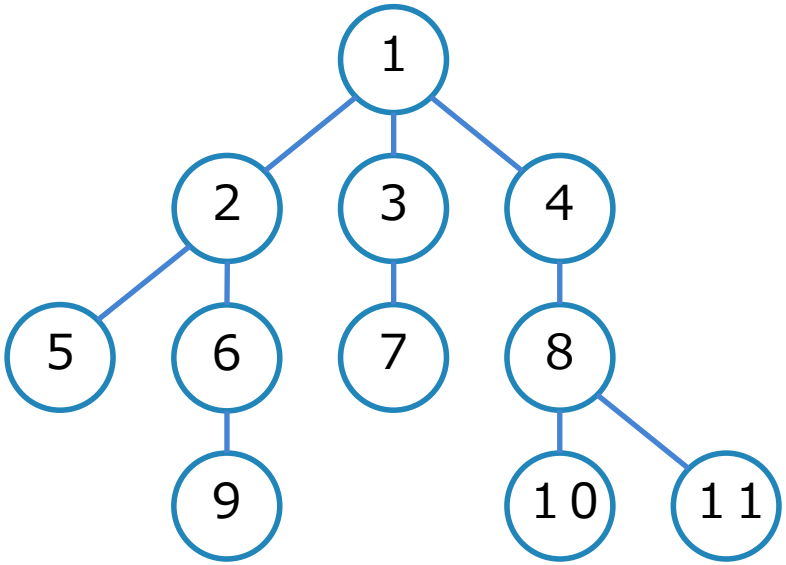
Take-Grant Model

Create

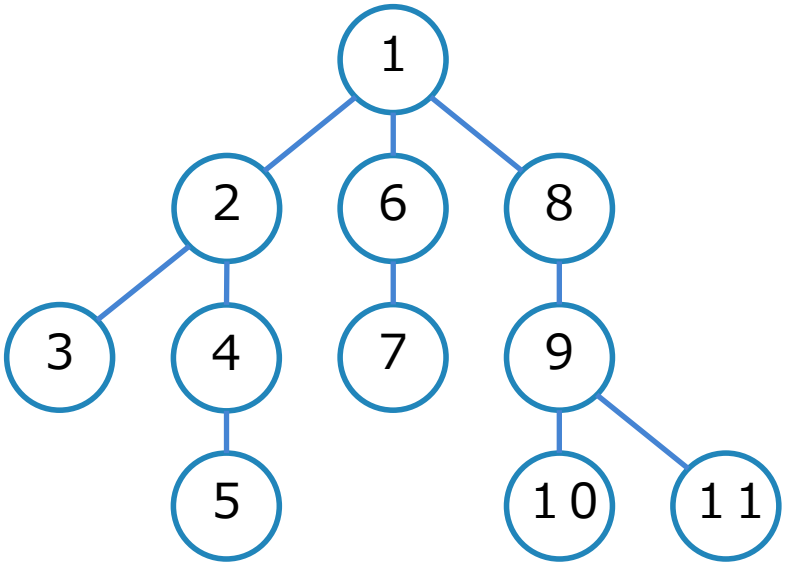


BFS(幅優先探索)

- 幅優先探索とは、木構造やグラフの探索を行うためのアルゴリズム
- 始点となるノードから隣接するノードを探索し、そこからさらに隣接するノードにたいして探索を繰り返して目的のノードを見つける

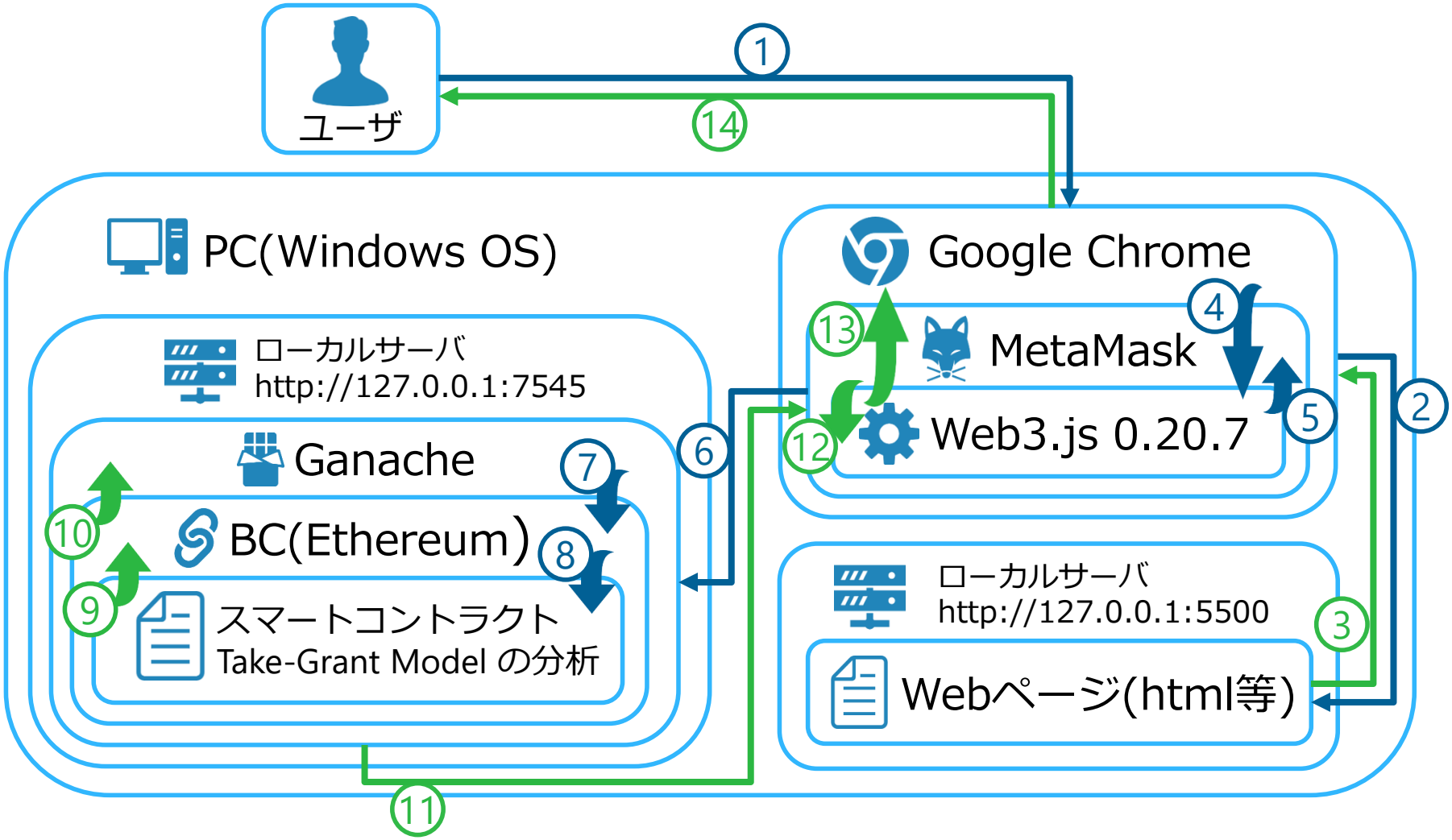


BFS（幅優先探索）

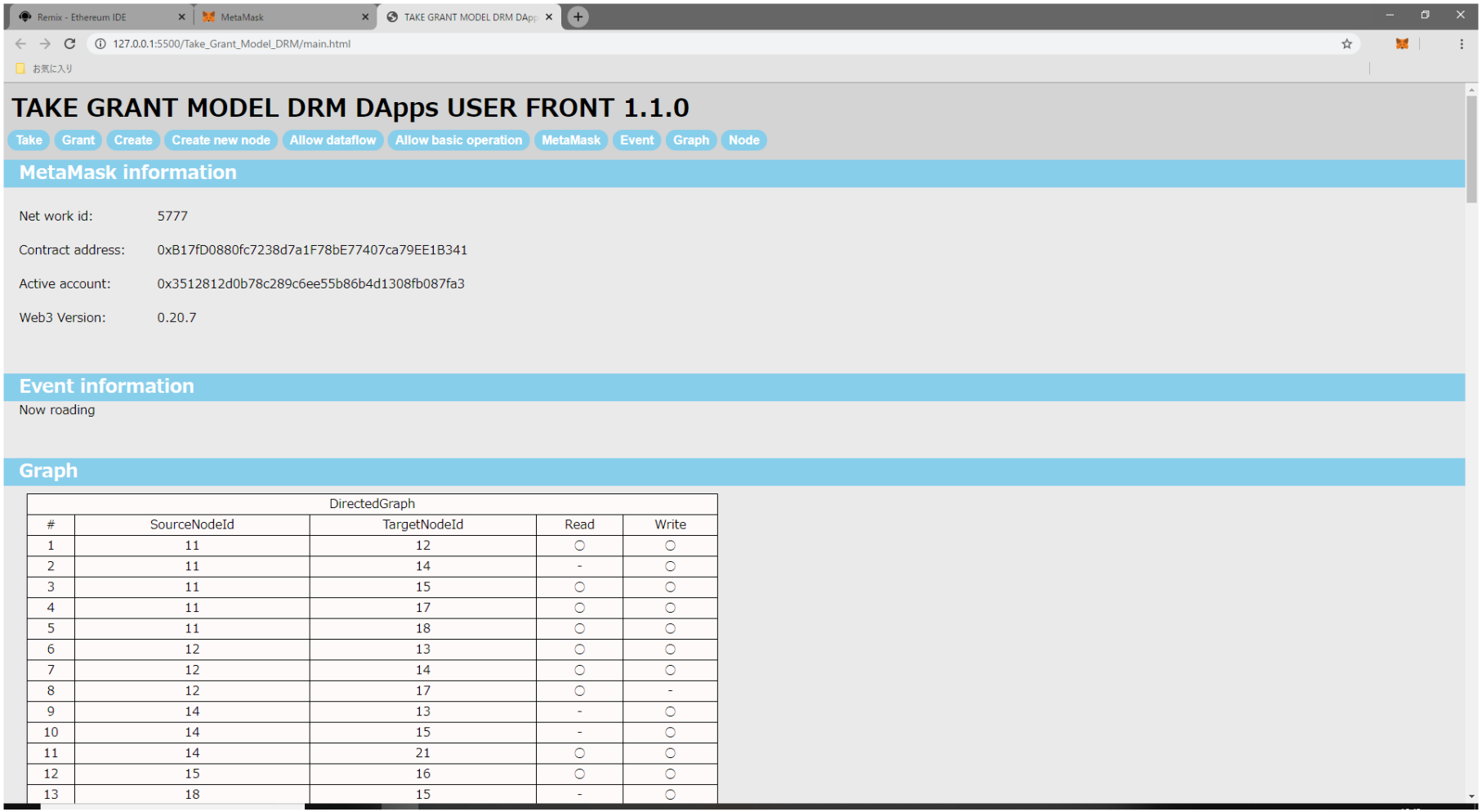


DFS（深さ優先探索）

実装システム概要 (予稿)


ユーザ
1
14
PC(Windows OS)
Google Chrome
4
13
MetaMask
5
2
12
Web3.js 0.20.7
ローカルサーバ
http://127.0.0.1:7545
6
Ganache
7
10
BC(Ethereum)
8
9
スマートコントラクト
Take-Grant Model の分析
ローカルサーバ
http://127.0.0.1:5500
3
Webページ(html等)
11

専用フロントエンド



DirectedGraph

#	SourceNodeId	TargetNodeId	Read	Write
1	11	12	○	○
2	11	14	-	○
3	11	15	○	○
4	11	17	○	○
5	11	18	○	○
6	12	13	○	○
7	12	14	○	○
8	12	17	○	-
9	14	13	-	○
10	14	15	-	○
11	14	21	○	○
12	15	16	○	○
13	18	15	-	○

神奈川大学　松尾 浩貴

専用フロントエンド

Graph

DirectedGraph				
#	SourceNodeId	TargetNodeId	Read	Write
1	11	12	○	○
2	11	14	-	○
3	11	15	○	○
4	11	17	○	○
5	11	18	○	○
6	12	13	○	○
7	12	14	○	○
8	12	17	○	-
9	14	13	-	○
10	14	15	-	○
11	14	21	○	○
12	15	16	○	○
13	18	15	-	○
14	18	17	-	○
15	18	19	○	○
16	18	20	○	-
17	19	20	○	○

神奈川大学　松尾 浩貴

専用フロントエンド

Node information

Node id 20 Send

NodeOwner
0xb69c4061c21994bc151df2f47b47f748d8c01106

DataFlowLimit

#	SourceNodeId	TargetNodeId	AllowType
1	20	19	Trust
2	20	11	Trust
3	20	12	Monitor
4	20	13	Monitor
5	20	14	Monitor
6	20	15	Monitor
7	20	16	Monitor
8	20	17	Monitor
9	20	18	Monitor
10	20	21	Monitor

ProtectionNode
20

TakeLimit

#	SourceNodeId	TargetNodeId	IntermediaryNodeId	Read	Write
1	18	20	19	○	-

GrantLimit

#	SourceNodeId	TargetNodeId	IntermediaryNodeId	Read	Write
1	AllNode	AllNode	AllNode	○	○

CreateLimit

#	Address
1	0xb69c4061c21994bc151df2f47b47f748d8c01106

神奈川大学　松尾 浩貴

専用フロントエンド

Create

Source node 22

Data type ○ Array ◉ Target node list ☑ Source node list ☑ Search list data

DataFlowLimit				
#	NodeId	SourceNodeId	TargetNodeId	AllowType
☐ 1	NewNode	AllNode	AllNode	Trust

TakeLimit						
#	NodeId	SourceNodeId	TargetNodeId	IntermediaryNodeId	Read	Write
☐ 1	NewNode	AllNode	AllNode	AllNode	○	○

GrantLimit						
#	NodeId	SourceNodeId	TargetNodeId	IntermediaryNodeId	Read	Write
☐ 1	NewNode	AllNode	AllNode	AllNode	○	○

CreateLimit		
#	NodeId	TargetAddres
☐ 1	NewNode	All address

専用フロントエンド

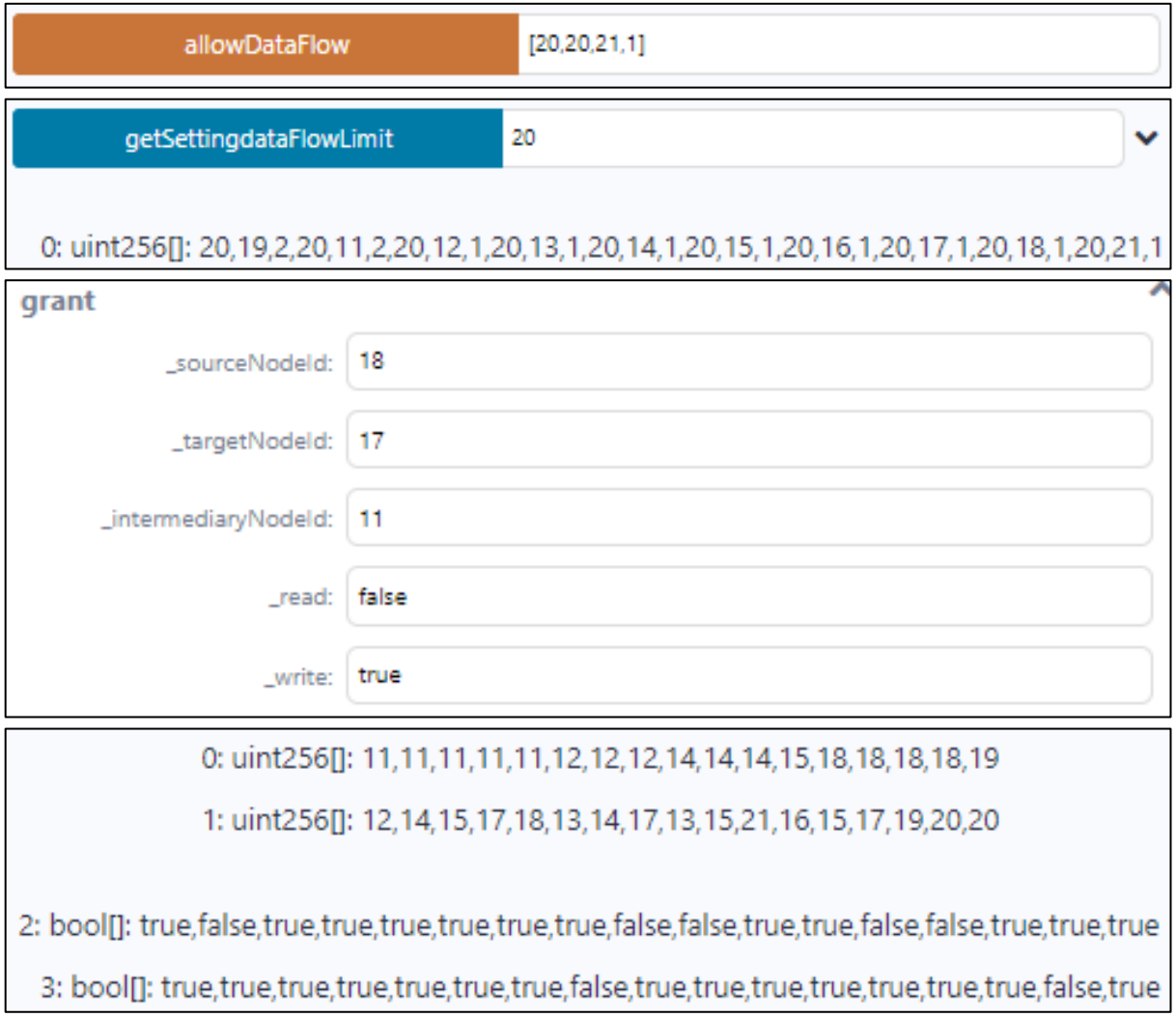
Event information

Event log

#	Log	
1	EventType:	createNewNode
	FromAddress:	0xb69c4061c21994bc151df2f47b47f748d8c01106
	NewNodeId:	22
	DataType:	TargetNodeList, SourceNodeList, SearchListData
	ValueForAllowDataFlow:	▼ In detail NodeId: 22, SourceNodeId: All node, TargetNodeId: All node, AllowType: Trust
	ValueForAllowBasicOperation:	▼ In detail BasicOperationType: Take, NodeId: 22, SourceNodeId: All node, TargetNodeId: All node, IntermediaryNodeId: All node, LabelType: Read, Write BasicOperationType: Grant, NodeId: 22, SourceNodeId: All node, TargetNodeId: All node, IntermediaryNodeId: All node, LabelType: Read, Write BasicOperationType: Create, NodeId: 22, TargetAddress: 0xb69c4061c21994bc151df2f47b47f748d8c01106

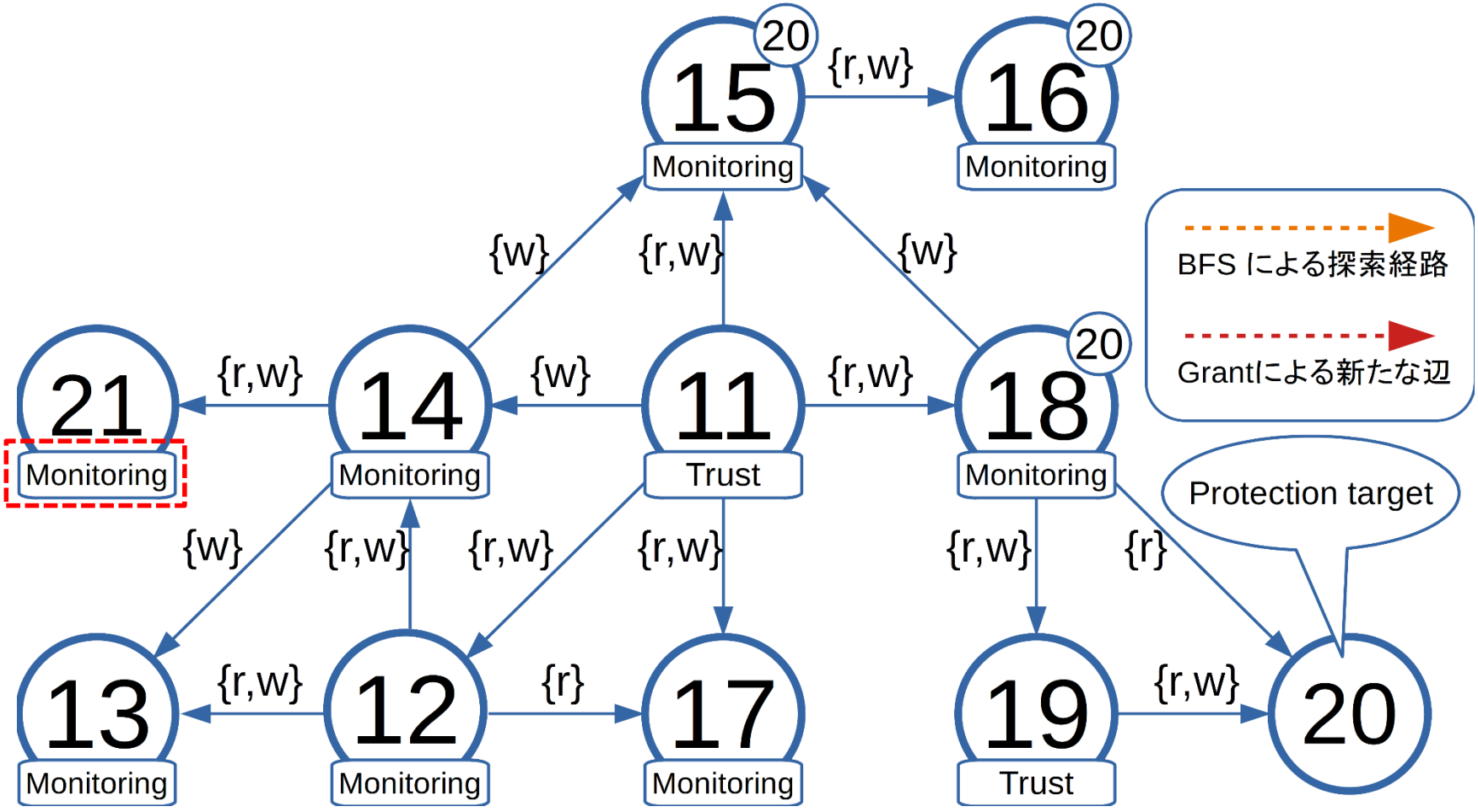
神奈川大学　松尾 浩貴

実験2



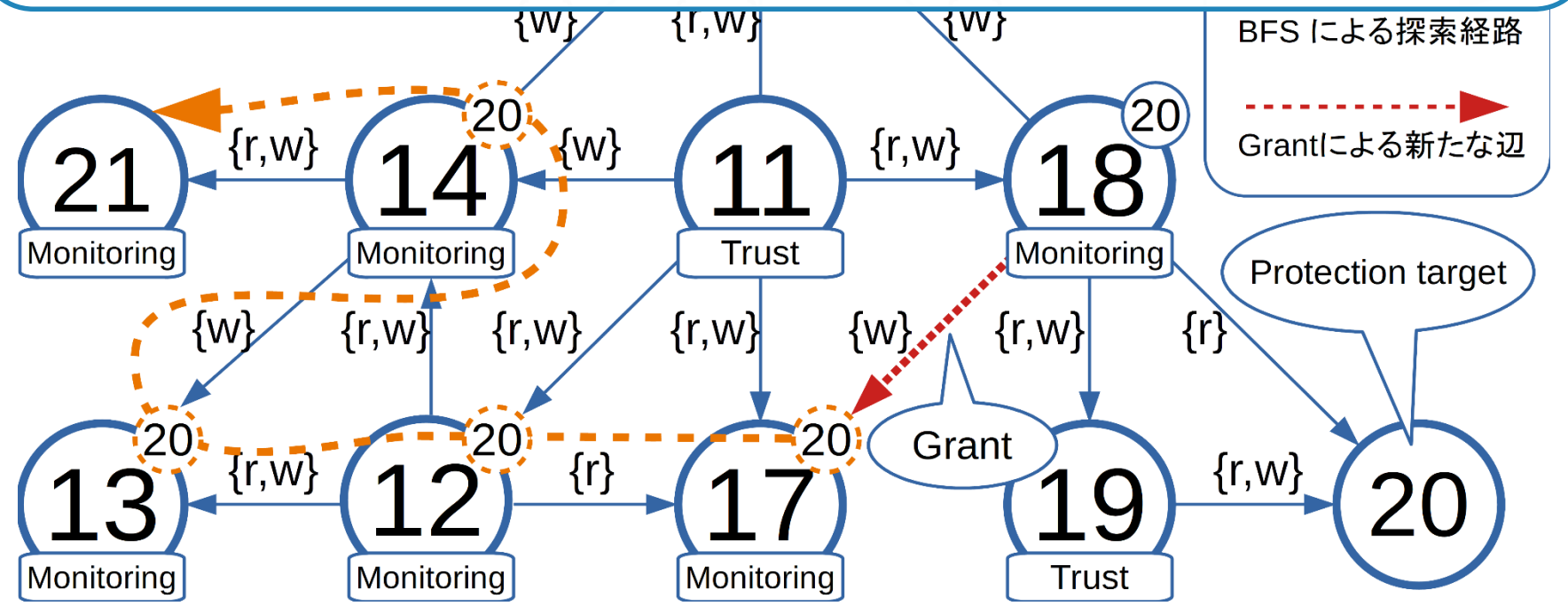
神奈川大学　松尾 浩貴

実験2

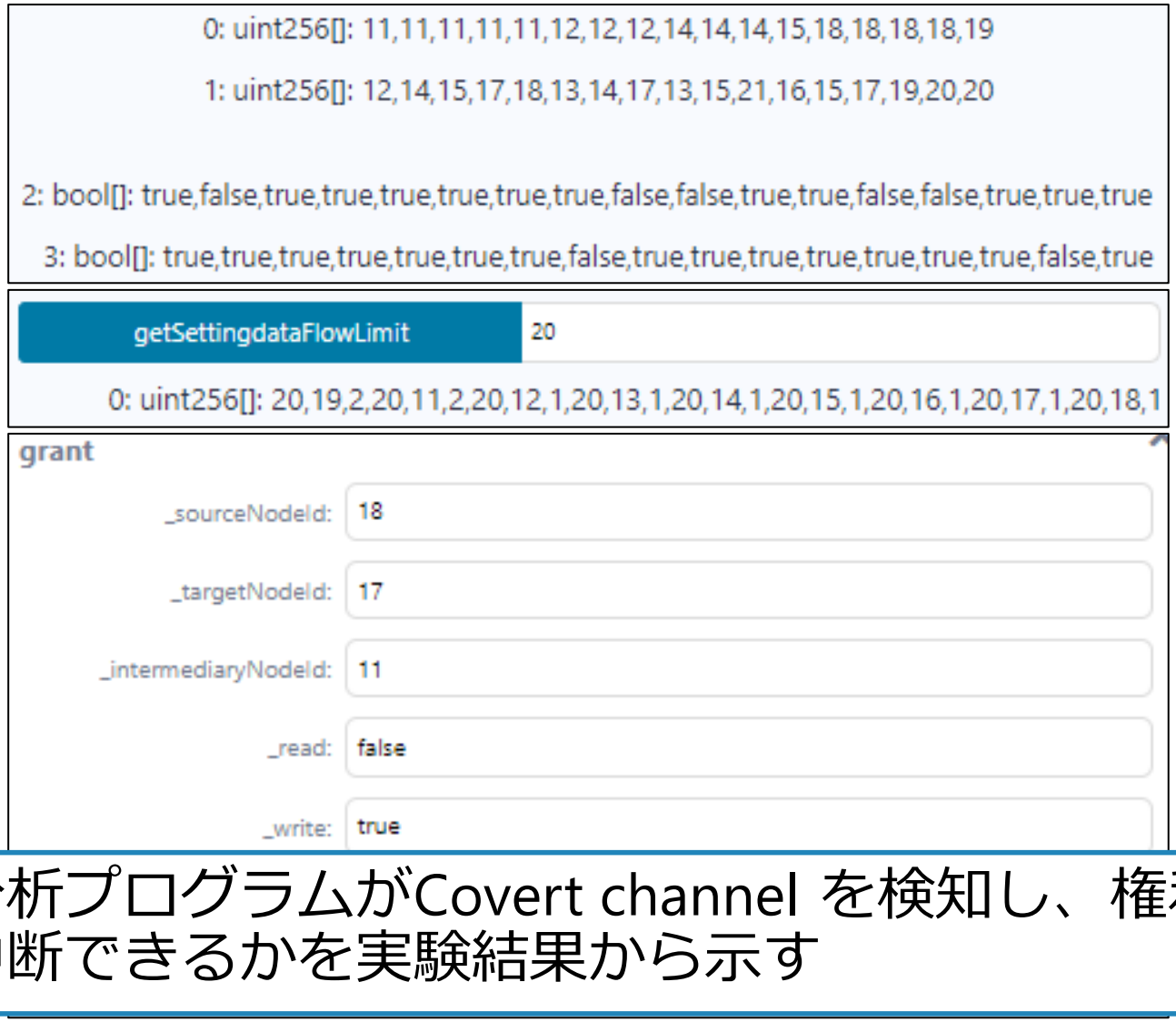

20
15
Monitoring
{r,w}
20
16
Monitoring
{w}
{r,w}
{w}
BFS による探索経路
Grantによる新たな辺
21
Monitoring
{r,w}
14
Monitoring
{w}
11
Trust
{r,w}
20
18
Monitoring
Protection target
{w}
{r,w}
{r,w}
{r,w}
{r,w}
{r}
13
Monitoring
{r,w}
12
Monitoring
{r}
17
Monitoring
19
Trust
{r,w}
20

実験2

ノード21は保護対象ノード20の課す制限がMonitoringに変更されることで、ノード20の情報を所持することを許され、探索において最終的に訪れるべきノードが無くなったため、Covert channel がないことが証明され基本操作が実行される

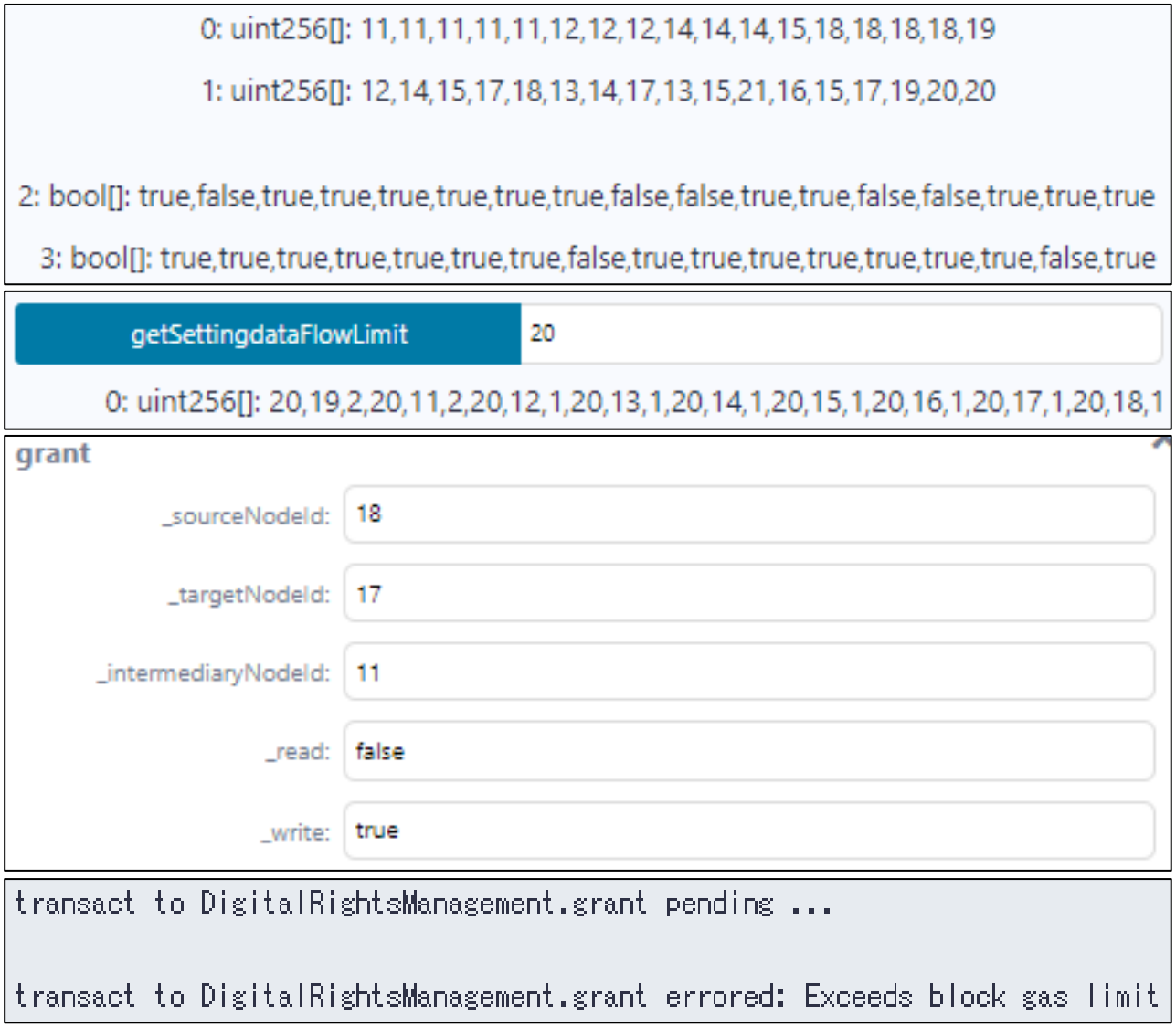


実験



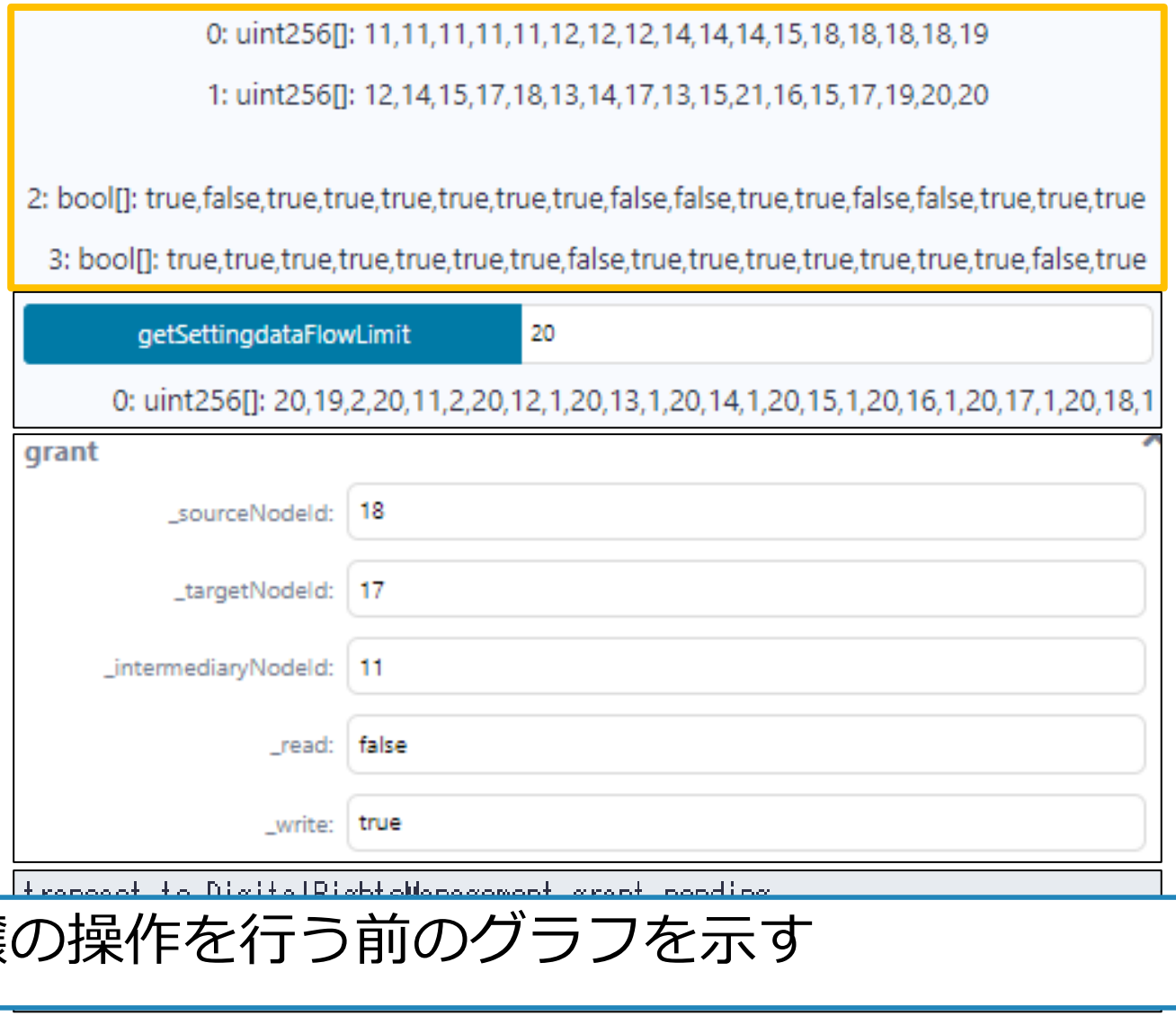
実際に分析プログラムがCovert channel を検知し、権利の委譲を中断できるかを実験結果から示す

実験



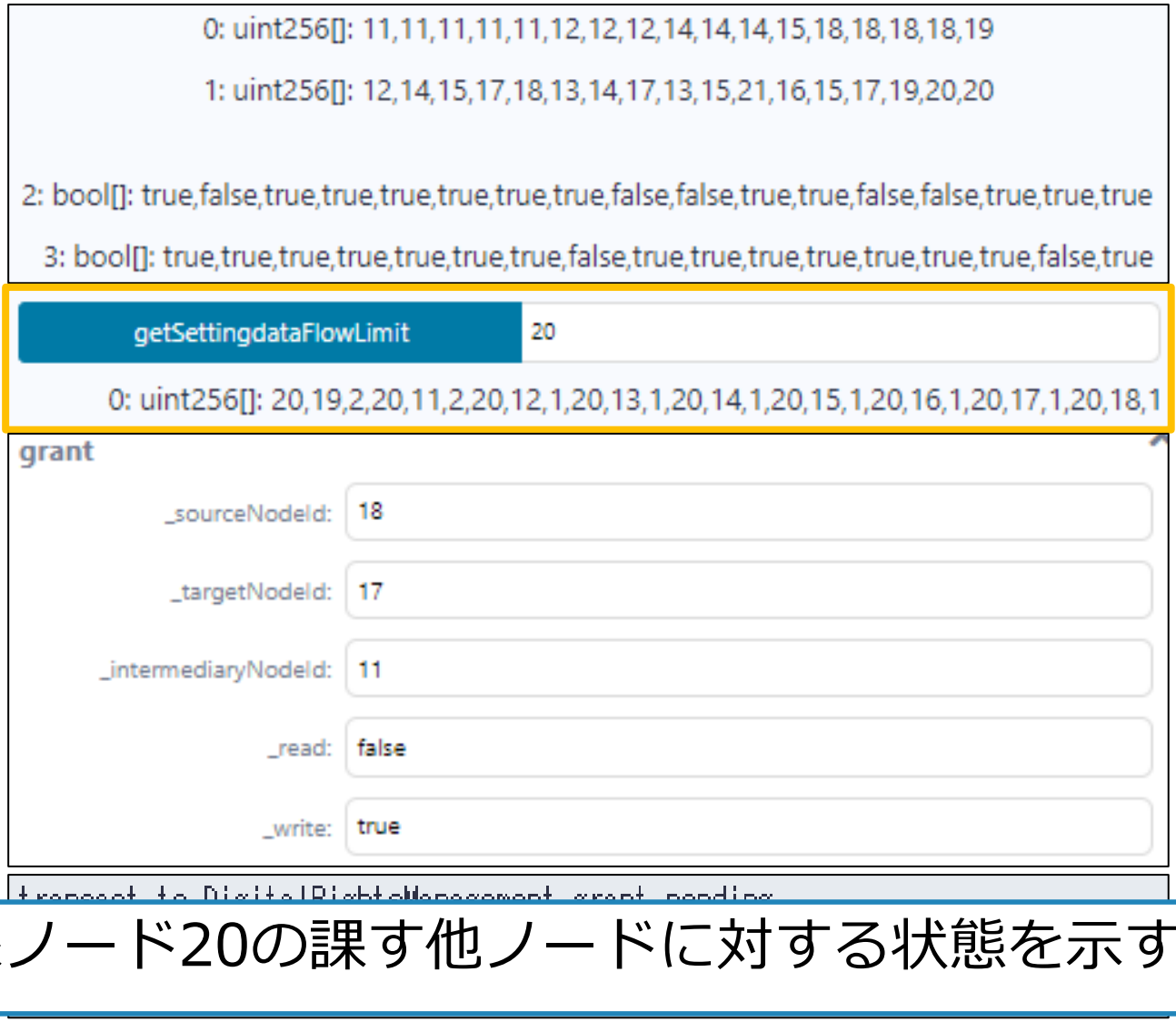
神奈川大学　松尾 浩貴

実験



権利委譲の操作を行う前のグラフを示す

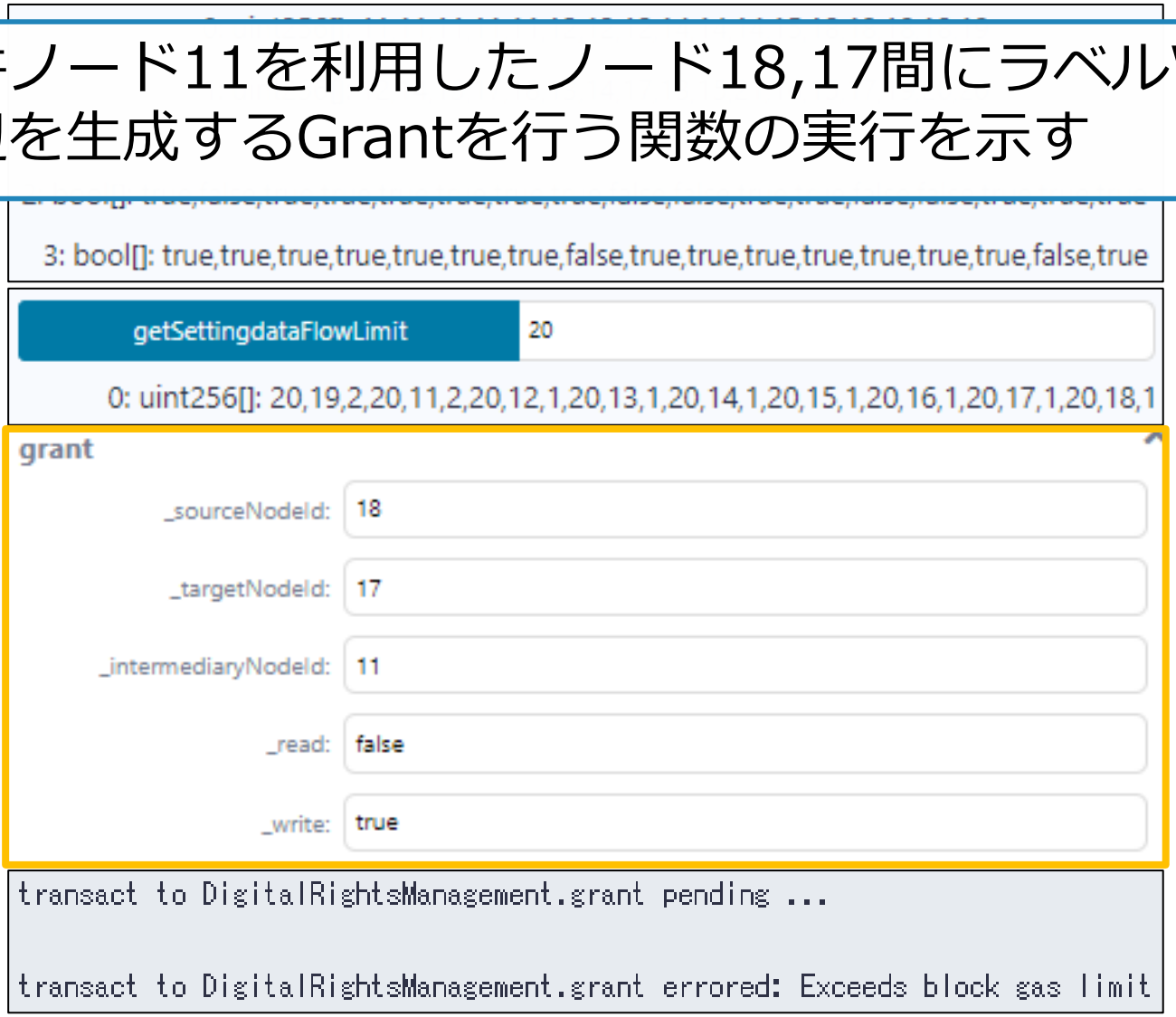
実験



保護対象ノード20の課す他ノードに対する状態を示す

実験

前提条件ノード11を利用したノード18,17間にラベルWriteの新たな辺を生成するGrantを行う関数の実行を示す



実験

関数を実行した結果、エラーが発生し、Grantの中断が行われたことを示す

神奈川大学　松尾 浩貴