Contrats autonomes : déploiement, interactions et utilisation d'oracles

Jérémy Toussaint
Pierre Jeanjacquot
Zied Guesmi

ressources: http://hack.iex.ec

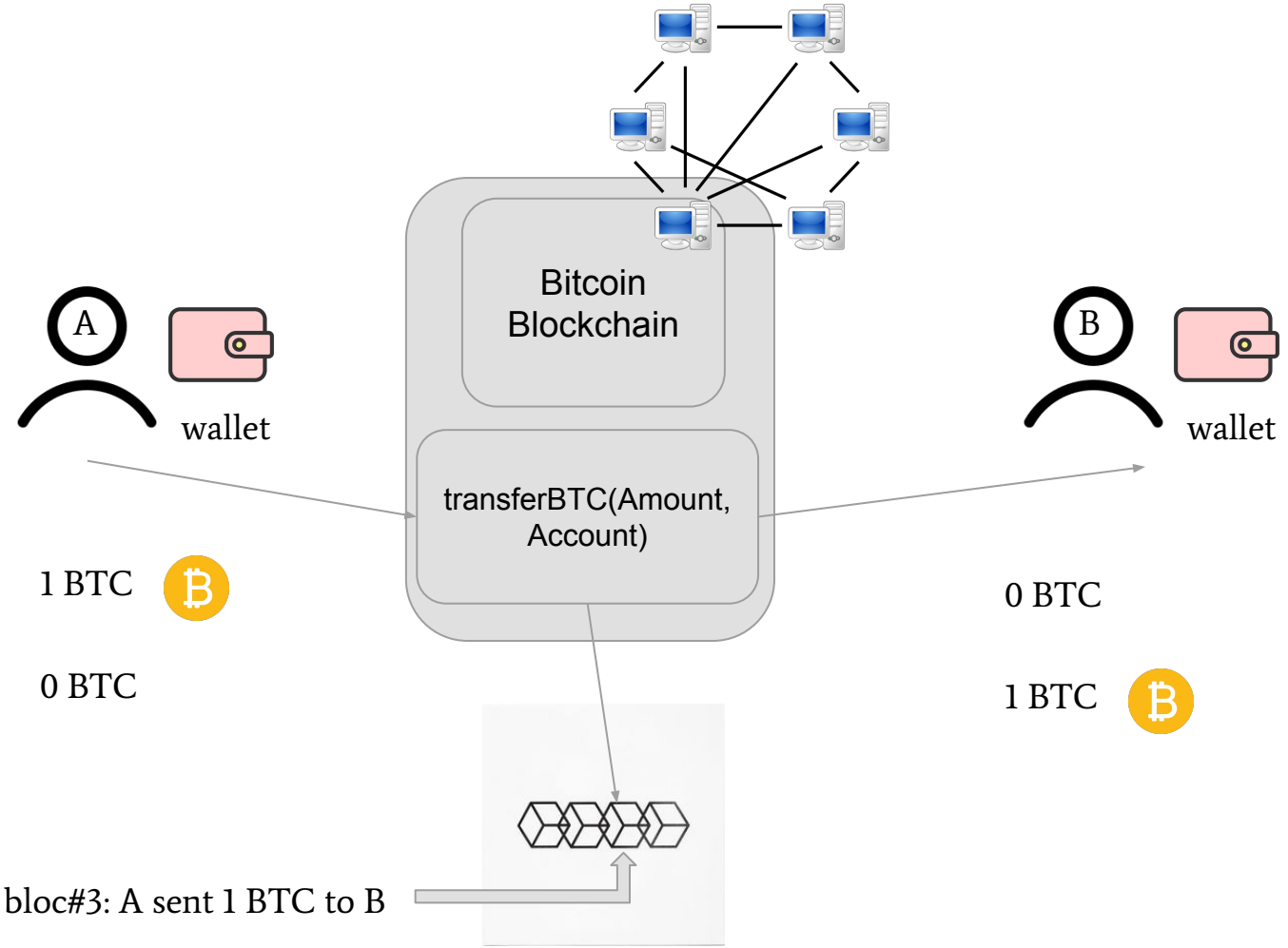
Déroulé

- Résumé des épisodes précédents
- La notion d'oracle (au sens blockchain)
- Live coding
  - Déploiement d'un smart contract
  - Interactions
- Analyse de Fizzy App : un service d'assurance qui utilise un oracle
- Importance d'utiliser des oracles "robustes"
- Brainstorming
- Questions et discussion

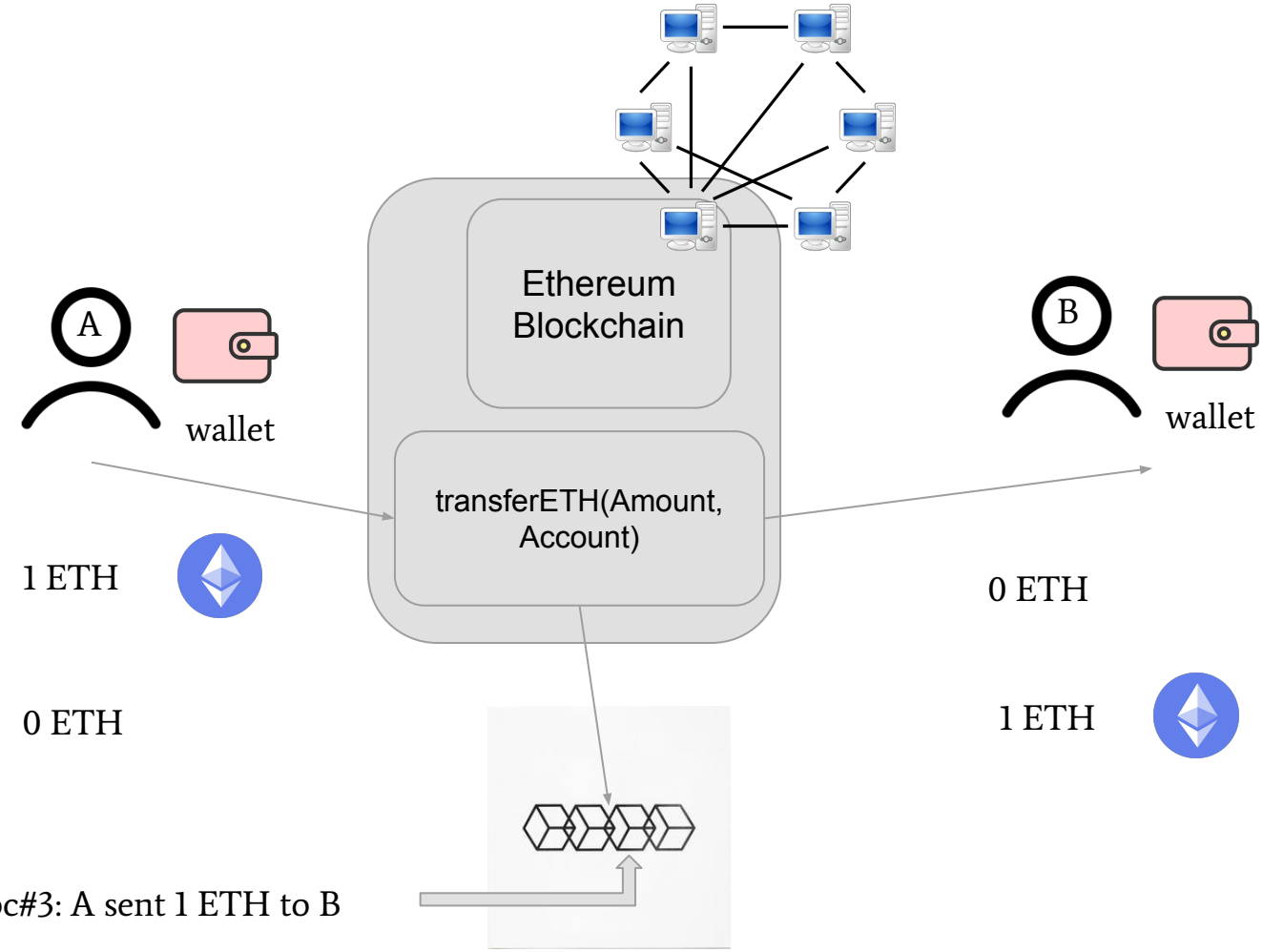
Résumé du Block 2

- Qu’est ce qu’une blockchain
  - Protocol
  - Transactions
  - Consensus et Création des blocks
- Qu’est ce qu’un wallet
  - Clé privée, clé publique et signature
  - Créer un wallet avec Metamask
  - Recevoir et envoyer des tokens
- Développer sur la blockchain
  - Qu’est-ce qu’un smart contract
  - Acteurs et cycle de vie
  - Exemple d’application

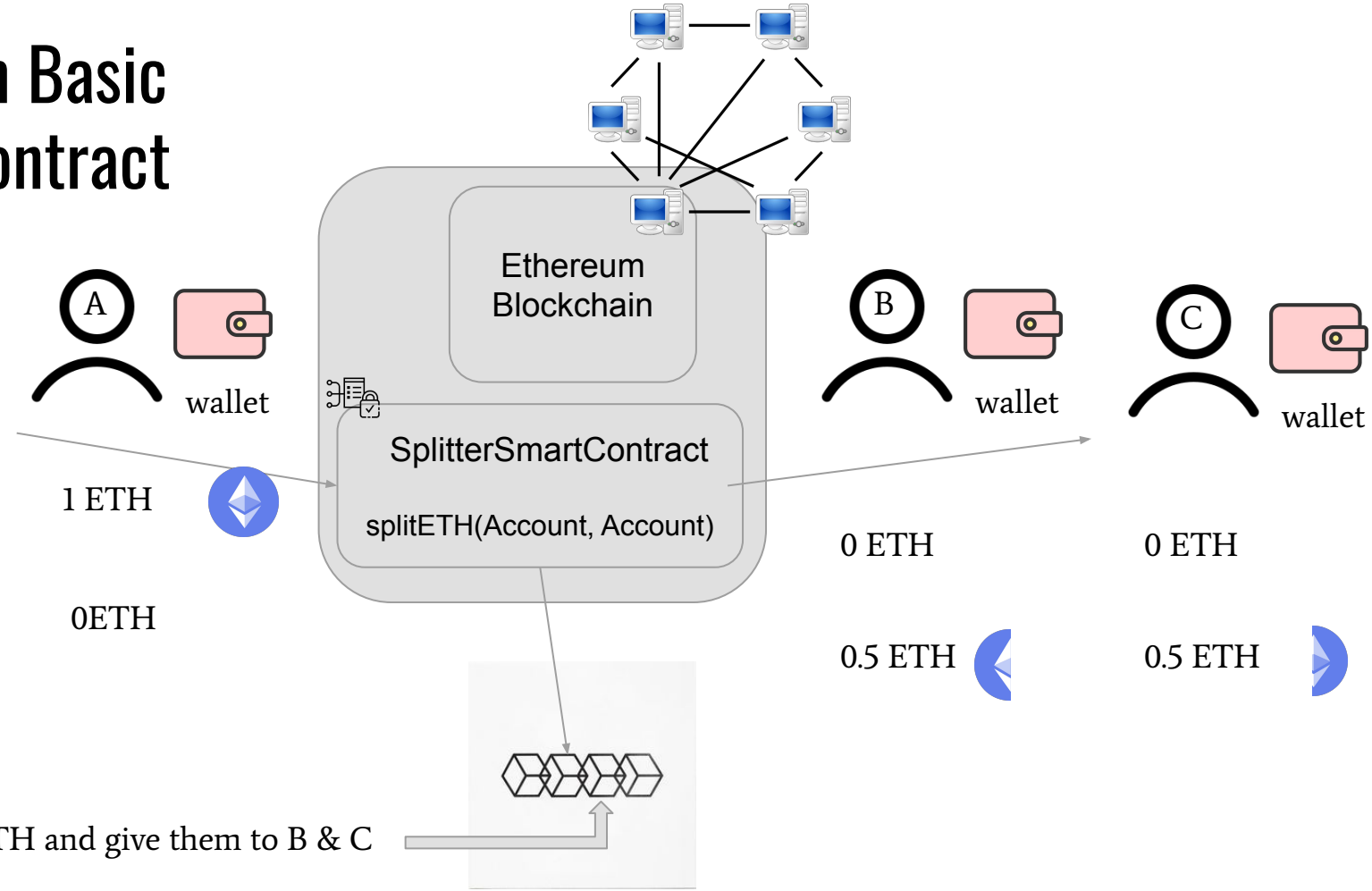
Bitcoin transfer



Ethereum Transfer

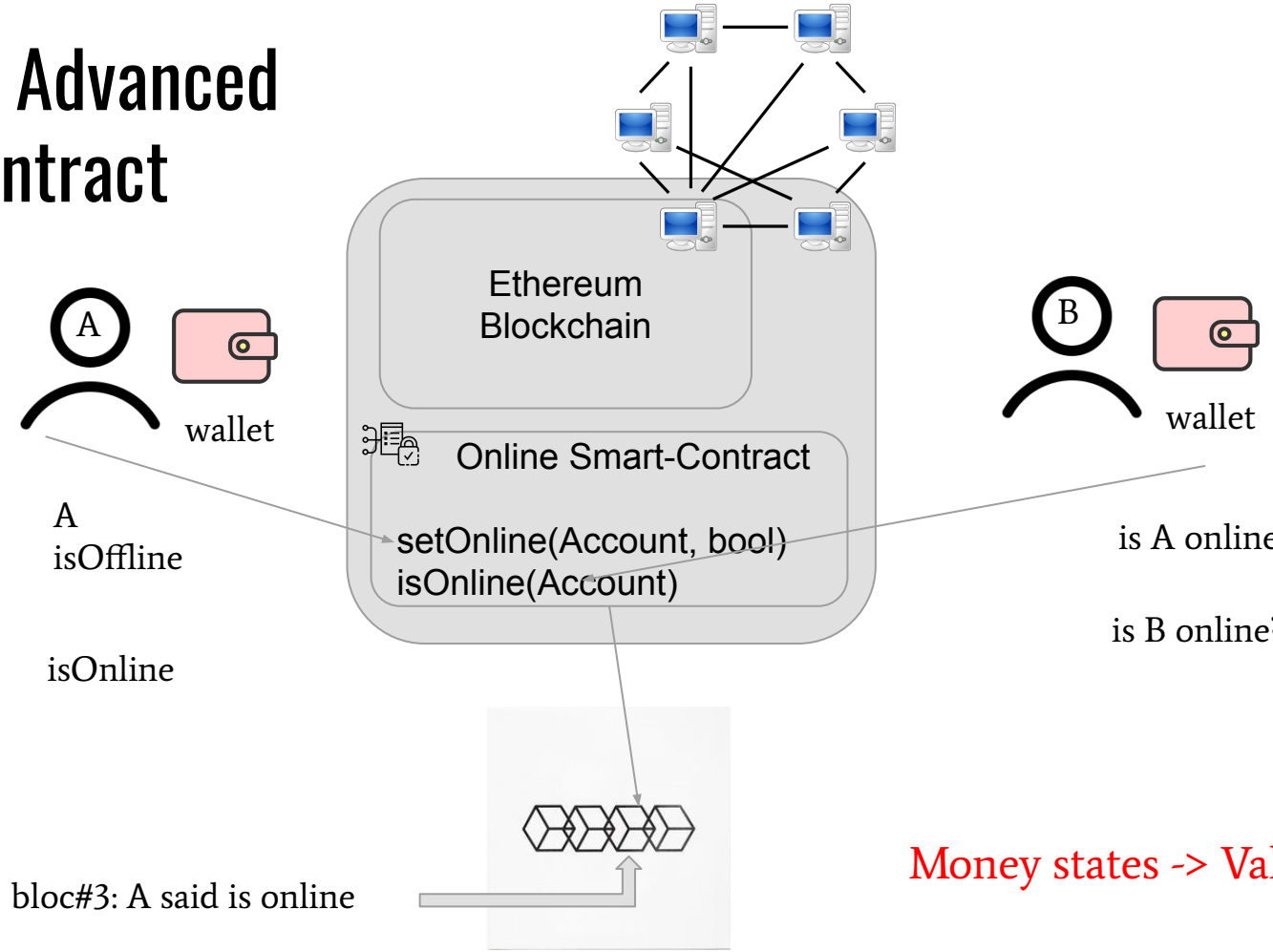


Ethereum Basic Smart-Contract



bloc#3: A split 1 ETH and give them to B & C

Ethereum Advanced Smart-Contract



Money states -> Value states

Oracle : Intro

Assurance Vacances sans pluie ™

BestInsurance

s'engage à verser une compensation financière à

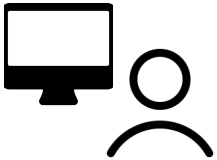
l'assuré

en cas de pluie durant son séjour à Palma de Majorque

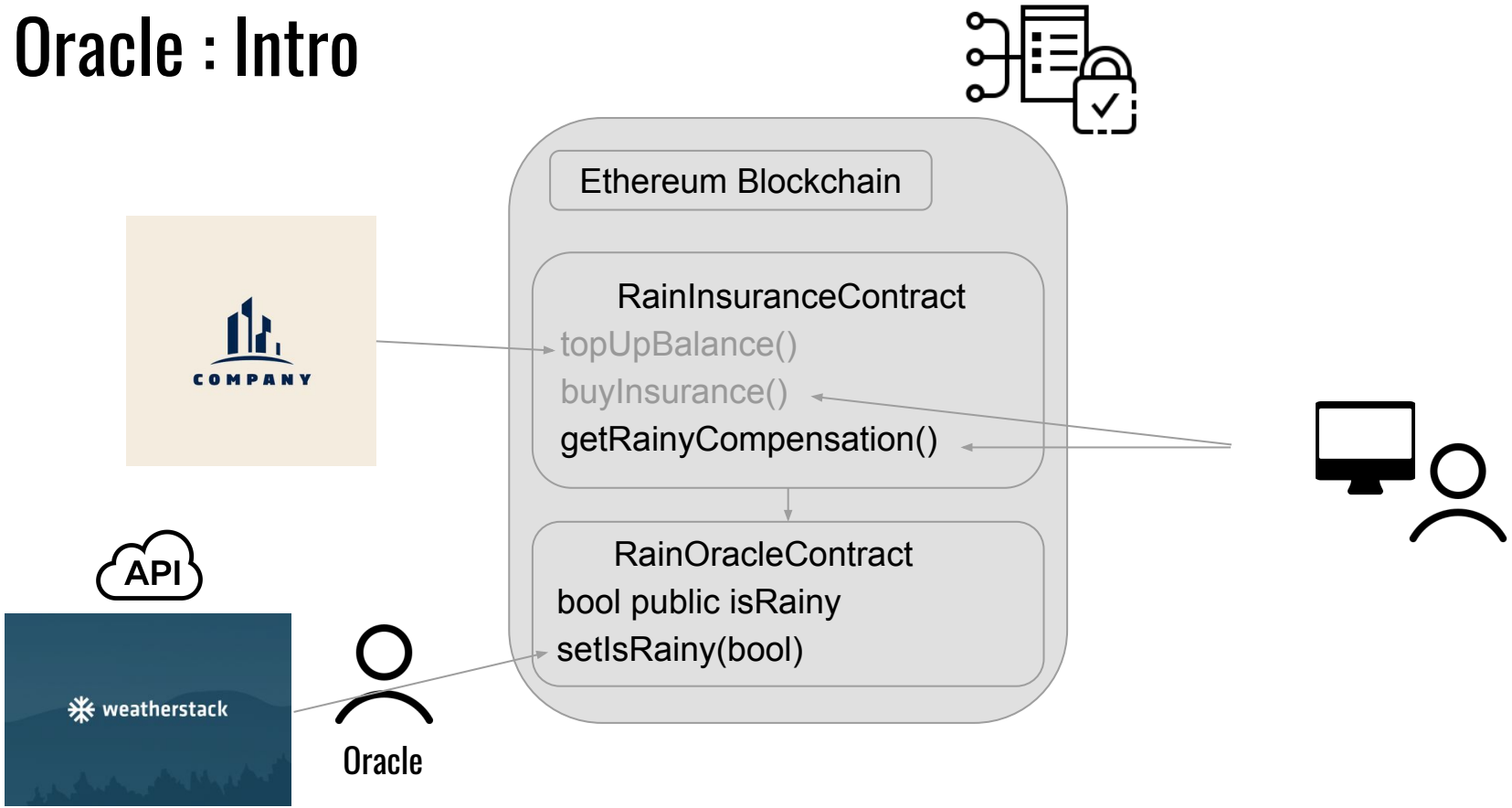
COMPANY

API

weatherstack



Oracle : Intro



http://api.weatherstack.com/current?access_key=xxx&query=palma

Let’s code!

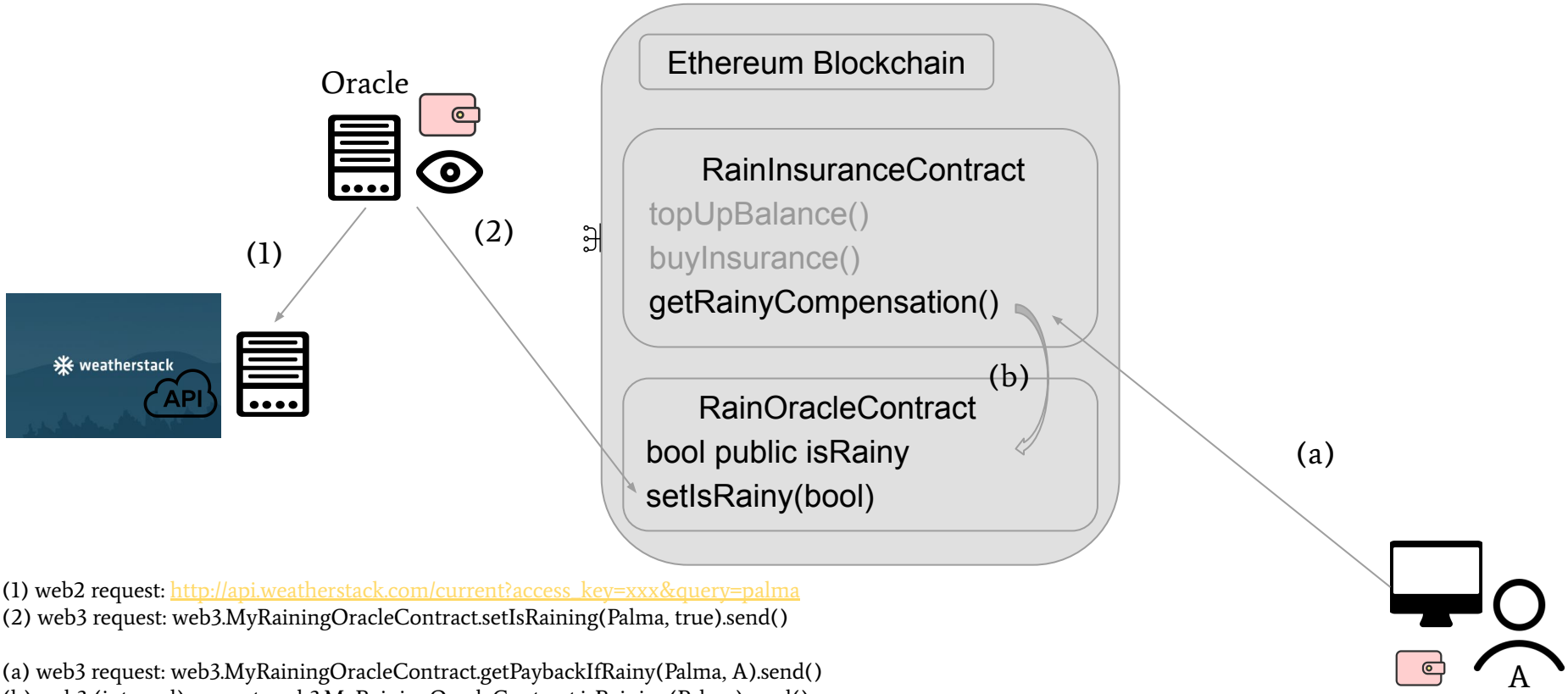
Utiliser l’IDE en ligne Remix: https://remix.ethereum.org

- coder un smart contract simple qui permet au client d’être indemnisé en cas de pluie
- déployer sur une chaîne de test locale (remix js VM)
- déployer sur la chaîne de test publique kovan avec Metamask

Interagir avec un smart contract via Etherscan: https://kovan.etherscan.io/

- vérifier les sources du smart contract déployé
- lire une valeur sur le smart contract
- exécuter une méthode du smart contract

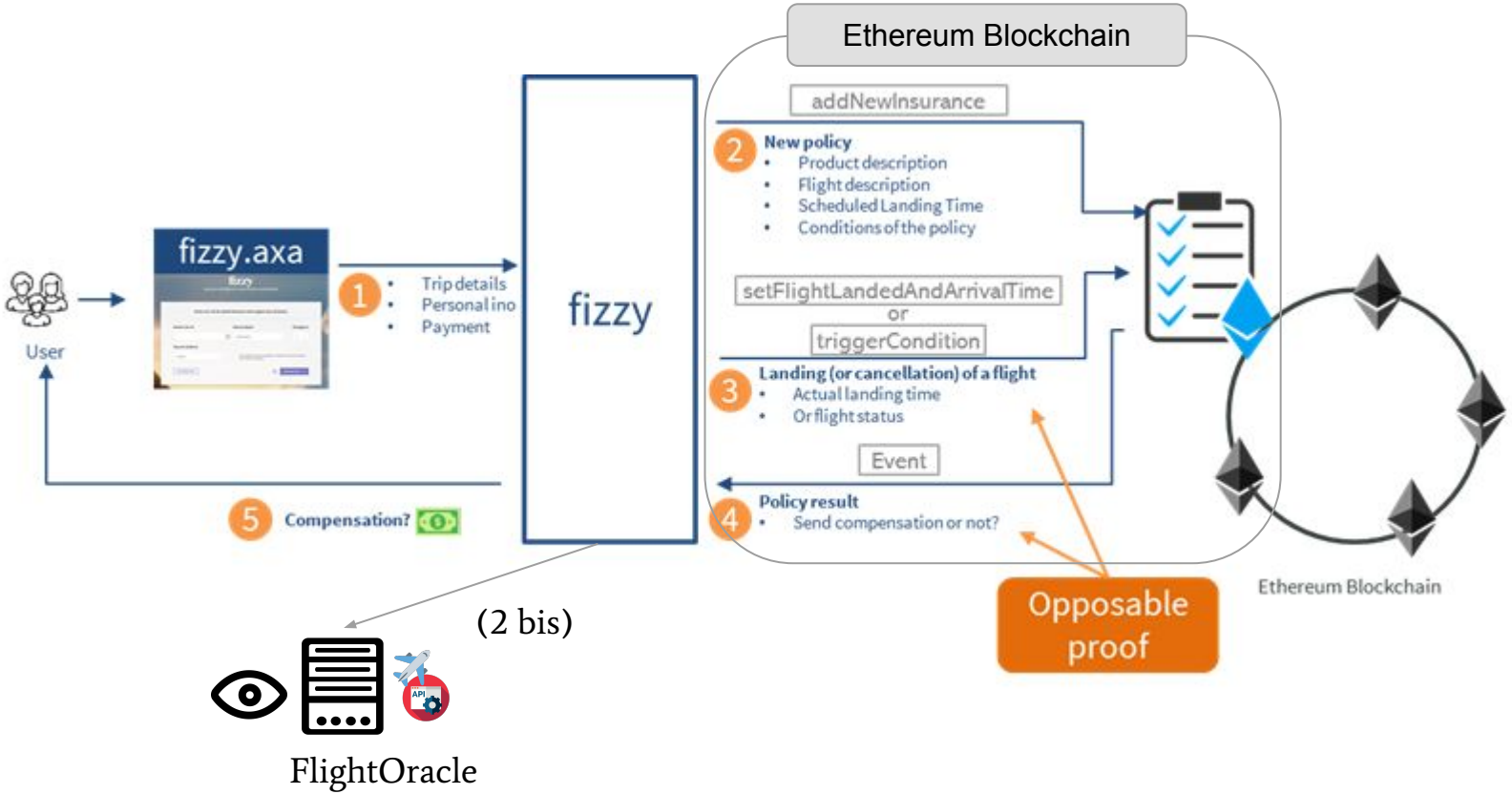
Oracle : le principe



(1) web2 request: http://api.weatherstack.com/current?access_key=xxx&query=palma
(2) web3 request: web3.MyRainingOracleContract.setIsRaining(Palma, true).send()

(a) web3 request: web3.MyRainingOracleContract.getPaybackIfRainy(Palma, A).send()
(b) web3 (internal) request: web3.MyRainingOracleContract.isRaining(Palma).send()

Fizzy App : analyse d'un service d'assurance on-chain

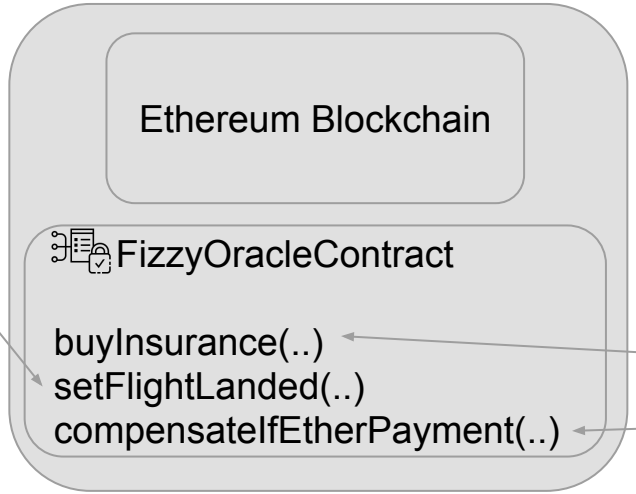


Fizzy App

https://etherscan.io/address/0xdc3d8fc2c41781b0259175bdc19516f7da11cba7#code

```
370     function setFlightLandedAndArrivalTime(
371         bytes32 flightId,
372         uint256 actualArrivalTime)
373         external
374         onlyOracle {
375
376         for (uint i = 0; i < insuranceList[flightId].length; i++) {
377             Insurance memory insurance = insuranceList[flightId][i];
378             if (insurance.status == InsuranceStatus.Open) {
379                 InsuranceStatus newStatus;
380                 uint256 triggeredCondition;
381
382                 if (_containsCondition(insurance.conditions, DELAY)) {
383                     if (actualArrivalTime > insurance.limitArrivalTime) {
384                         triggeredCondition = DELAY;
385                         newStatus = InsuranceStatus.ClosedCompensated;
386                         compensateIfEtherPayment(insurance);
387                     } else {
388                         triggeredCondition = NONE;
389                         newStatus = InsuranceStatus.ClosedNotCompensated;
390                         noCompensateIfEtherPayment(insurance);
391                     }
392                 } else {
393                     triggeredCondition = NONE;
394                     newStatus = InsuranceStatus.ClosedNotCompensated;
395                     noCompensateIfEtherPayment(insurance);
396                 }
397
398                 insuranceList[flightId][i].status = newStatus;
399
400                 emit InsuranceUpdate(
401                     flightId,
402                     insurance.productId,
403                     insurance.premium,
404                     insurance.indemnity,
405                     triggeredCondition,
406                     newStatus
407                     );
408             }
409         }
410     }
```

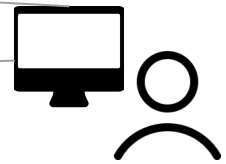
(3)




```
631     function buyInsurance(
632         bytes32         flightId,
633         uint256         productId,
634         uint256         premium,
635         uint256         indemnity,
636         uint256         taxes,
637         uint256         limitArrivalTime,
638         uint256         conditions,
639         uint256         timestampLimit,
640         address         buyerAddress,
641         bytes calldata signature
642     )
```

(1)

(2)



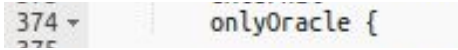
```
530     function compensateIfEtherPayment(Insurance memory insurance) private {
531         if (insurance.compensationAddress != address(0)) {
532             _compensate(insurance.compensationAddress, insurance.indemnity, insurance.productId);
533         }
534     }
```

Limit: Centralized Oracle

Oracle : Faire confiance à l'oracle

- Niveau 0 : Qui a le droit de d'entrer les informations de l'oracle ?

    -> restriction d'accès au méthodes d'écriture

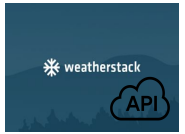


- Niveau 1 : L'entité qui entre les informations est-elle fiable ?

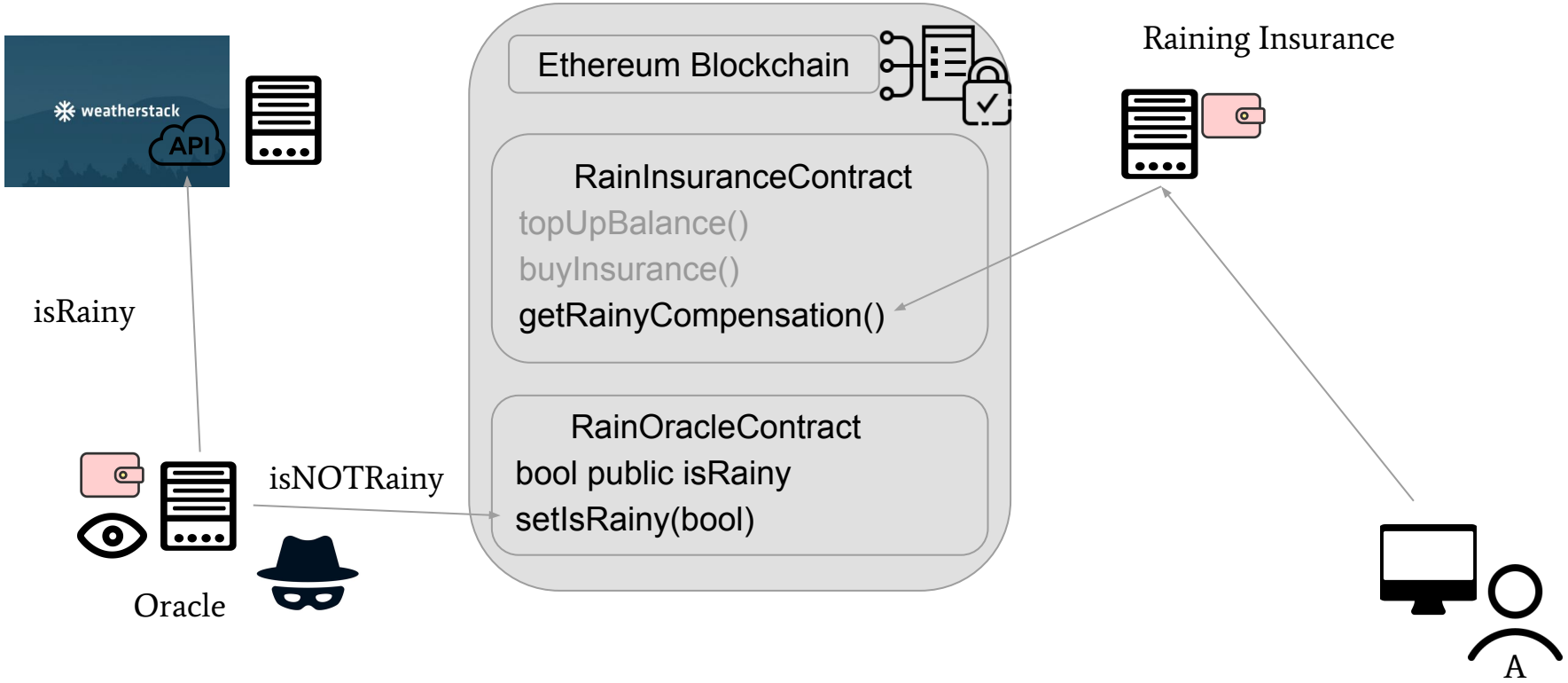
    -> utilisation d'un consensus

- Niveau 2 : La source de donnée est-elle fiable ?

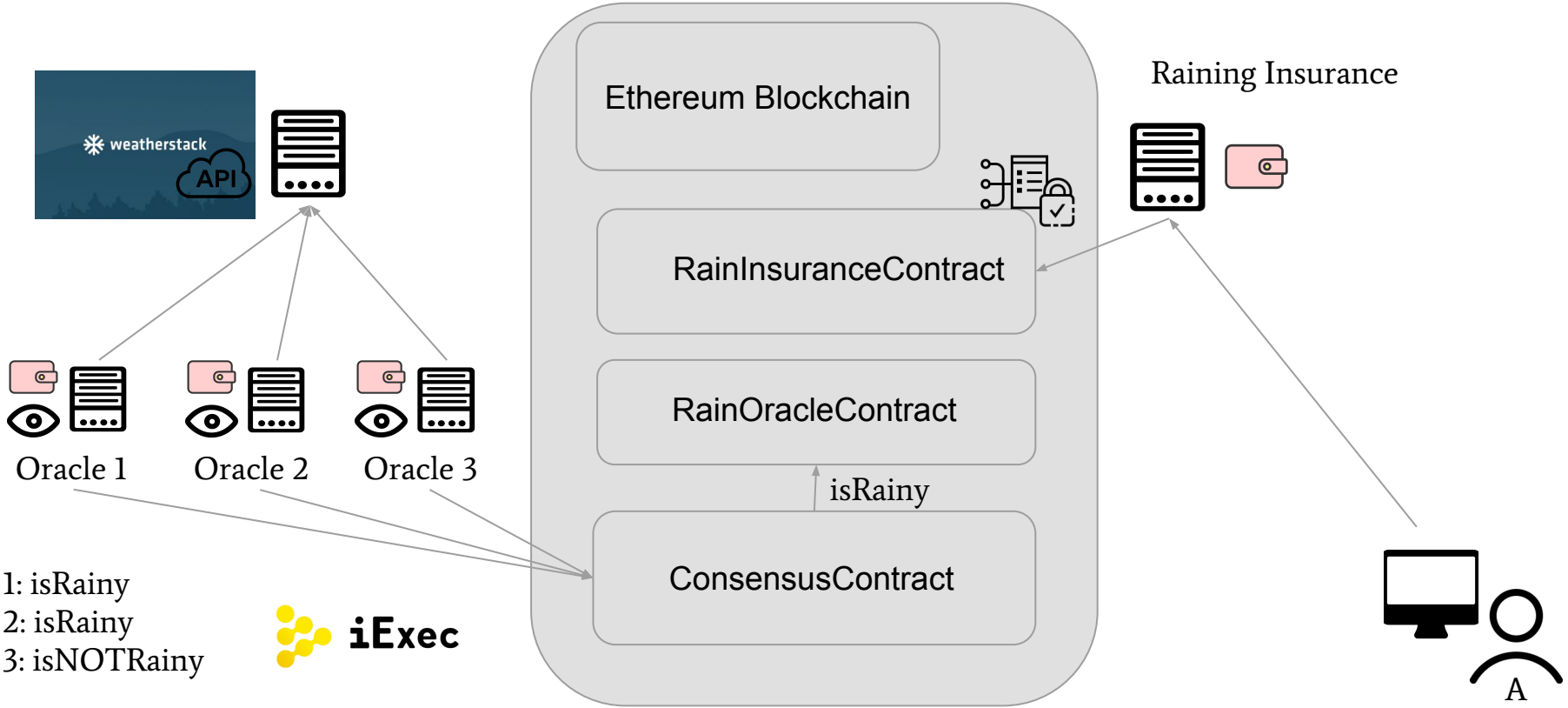
    -> croisement des informations de plusieurs sources



Centralized Oracle



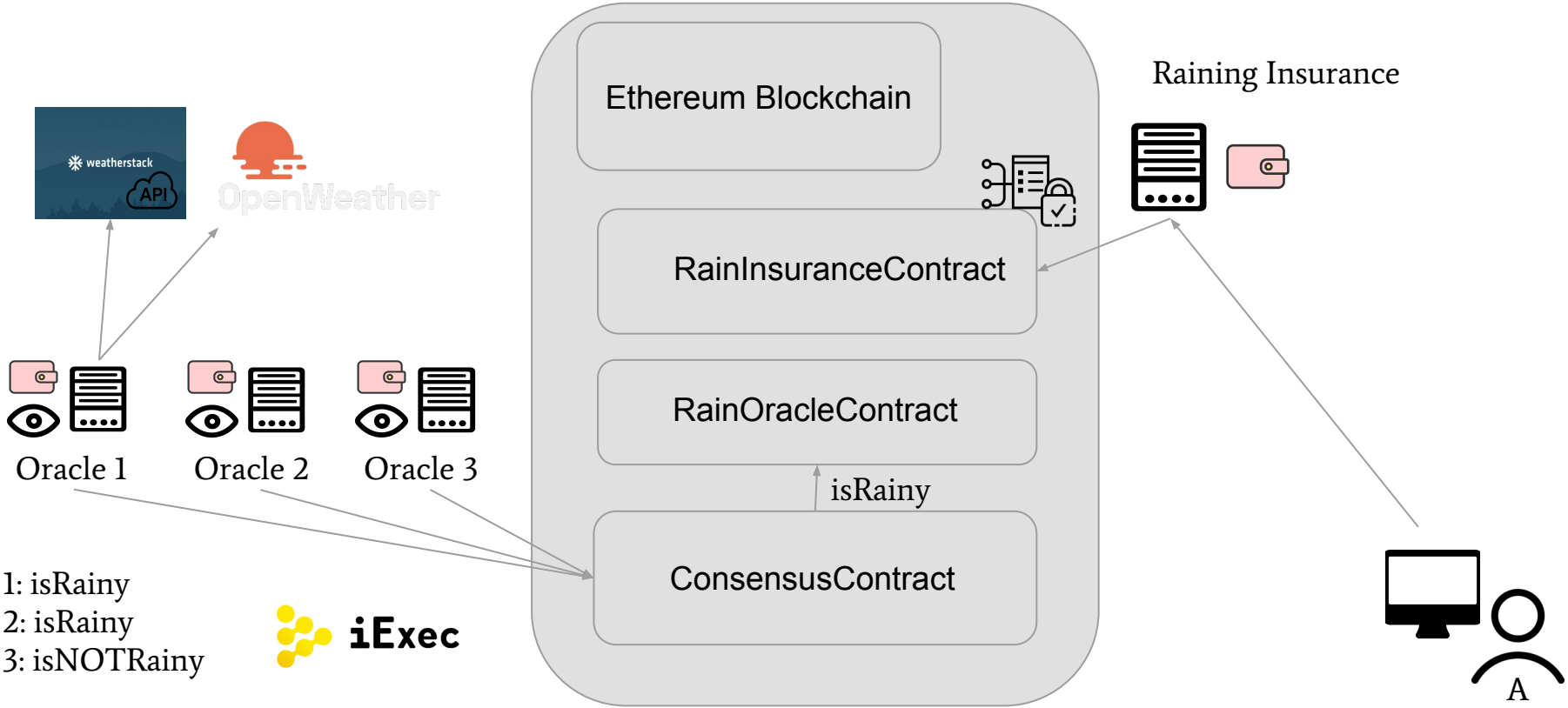
Decentralized Oracle: Level 1



1: isRainy
2: isRainy
3: isNOTRainy

iExec

Decentralized Oracle: Level 2



1: isRainy
2: isRainy
3: isNOTRainy

iExec

Les oracles “on top of” iExec

- Taux de change de monnaies (price feed)
- Fournisseur d’aléatoire (Random Number Generator)
- Le temps qu’il fait
- Anémomètre (pour mesurer le vent)
- Résultats de foot
- Suivi de coli
- Processeur de paiement (VISA)
- Bitcoin tx verifier
- Info réseau traffic aérien (flight delay, ...)
- Certification de contenu de page web (TLS Notary)

Brainstorming Session

- Exemple 1 : “micro-assurance anti-grêle pour les agriculteurs”
- Exemple 2 : “XXXXXXXXX”

À vous de jouer !