AWS re:Invent

NOV. 29 – DEC. 3, 2021 | LAS VEGAS, NV

BLC304-R2

Making legal contracts smarter

Rafia Tapia
Blockchain Specialist Solutions Architect
AWS

Dan Blaner
Sr. QLDB Specialist Solutions Architect
AWS

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

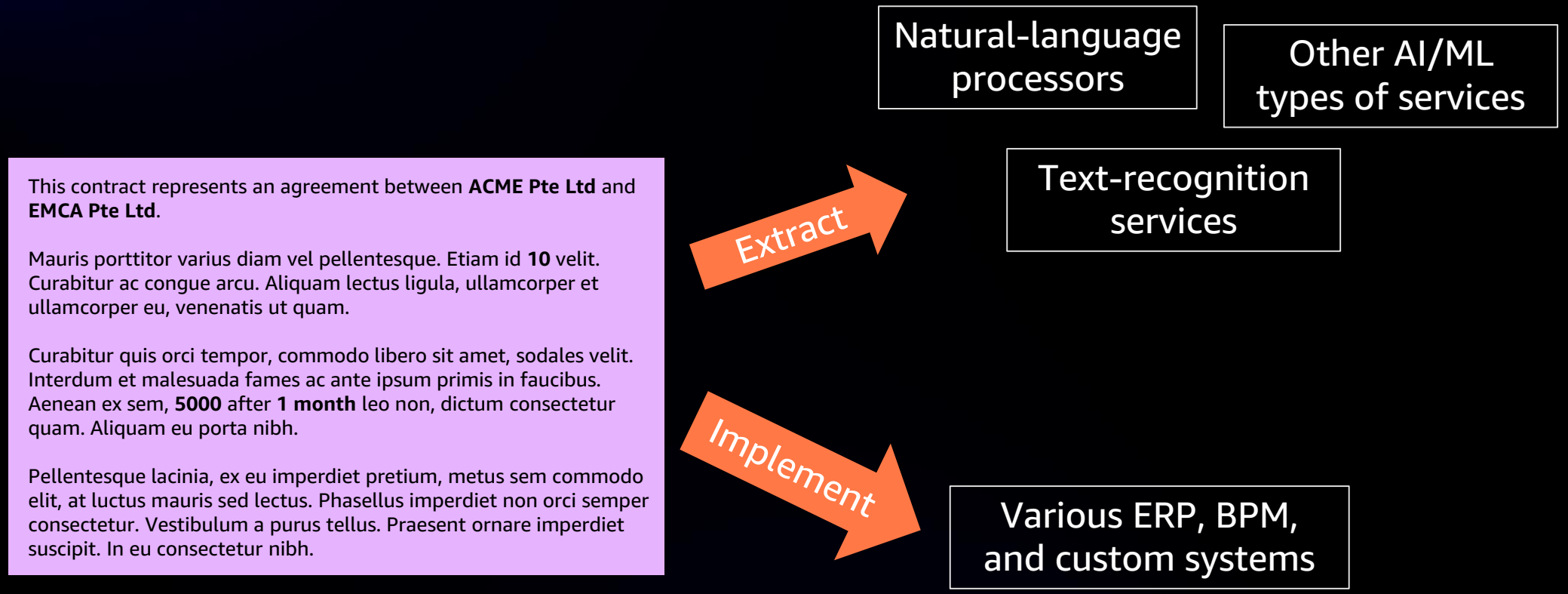
What is a smart legal contract?

Introduction to the Accord Project

Implementing smart legal contracts in Amazon QLDB

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Digital contracts are not smart



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Smart legal contracts

TURN UNSTRUCTURED BUSINESS DATA INTO A STRUCTURED FORM

This contract represents an agreement between **ACME Pte Ltd** and **EMCA Pte Ltd**.

Mauris porttitor varius diam vel pellentesque. Etiam id **10** velit. Curabitur ac congue arcu. Aliquam lectus ligula, ullamcorper et ullamcorper eu, venenatis ut quam.

Curabitur quis orci tempor, commodo libero sit amet, sodales velit. Interdum et malesuada fames ac ante ipsum primis in faucibus. Aenean ex sem **5000** after **1 month** leo non, dictum consectetur quam. Aliquam eu porta nibh.

Pellentesque lacinia, ex eu imperdiet pretium, metus sem commodo elit, at luctus mauris sed lectus. Phasellus imperdiet non orci semper consectetur. Vestibulum a purus tellus. Praesent ornare imperdiet suscipit. In eu consectetur nibh.



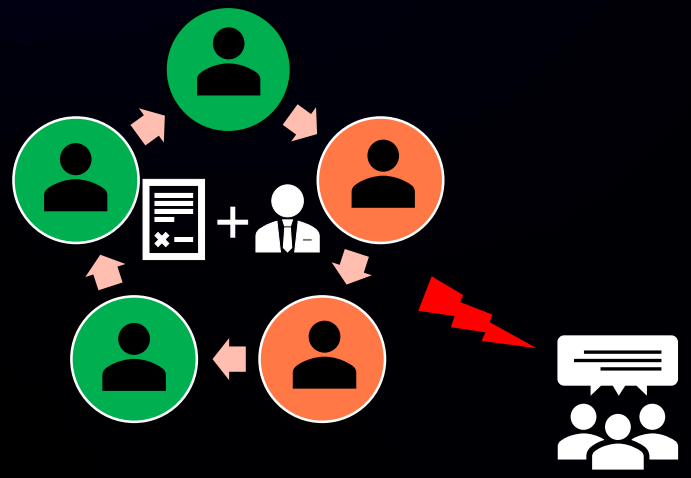
Extract business data using a contract template grammar and schema

Buyer	Seller	Quantity	Price	Duration
”ACME Pte Ltd”	“EMCA Pte Ltd”	10	5000	1, “month”

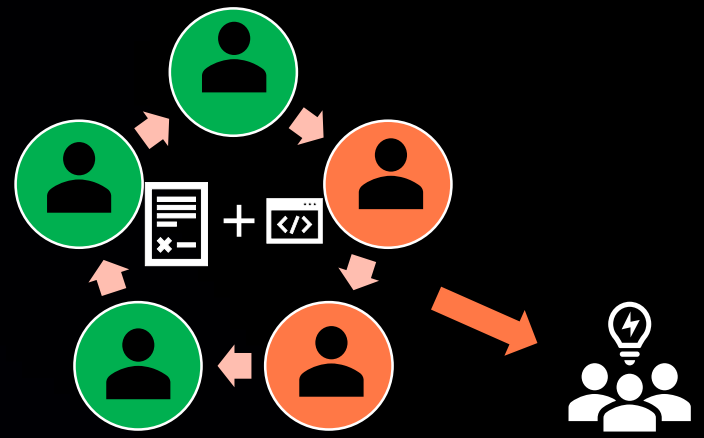
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Smart legal contracts

AUTOMATE CONTRACT CLAUSES



VS.



Legal contracts

- B2B legal contract clause automation is scattered across multiple systems, and verification is done by humans
- Differences in interpretations lowers trust and slows down the process

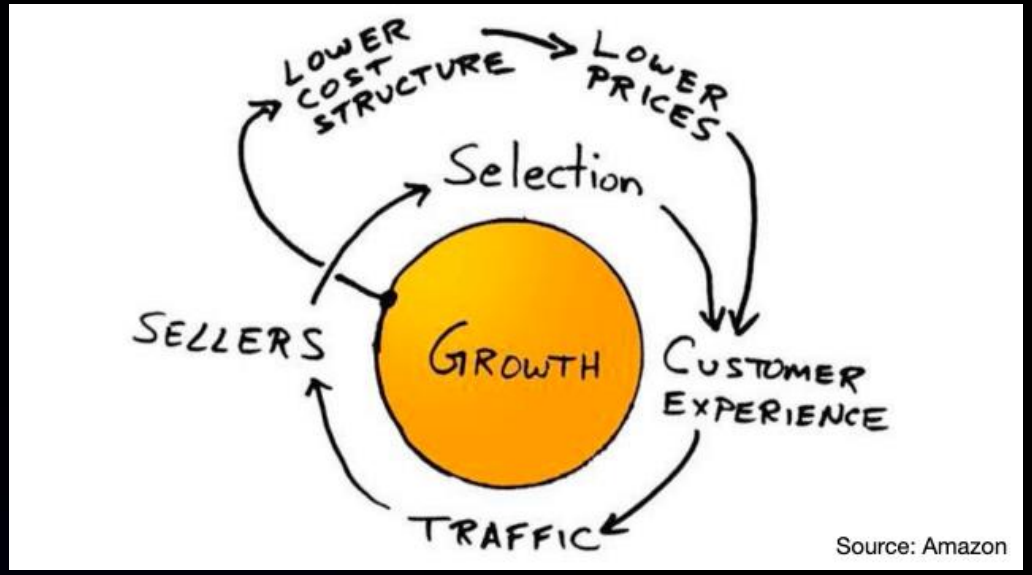
Smart legal contracts

- B2B legal contract verification is performed by a script
- Formal logic unifies interpretation; trust is higher, making process execution faster

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Benefits of using smart legal contracts

- Simplify the extraction, validation, and analysis of business data in legal contracts
- Increasing trust between business parties is a key to:
  - Increasing the speed of B2B processes
  - Switching to new and more efficient business models
  - Scaling up the flywheel effect



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Introducing the Accord Project

Provides a set of tools to develop and use smart legal contracts

- A templating system for smart legal contracts
- A library of UI components to construct web applications
- Contract designer and developer tools
- A catalog of open-source smart legal contract templates for a wide variety of use cases

Fully open source under Apache-2 license
http://www.accordproject.org

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Introducing the Accord Project

1. IDENTIFY KEY CONTRACT DATA IN THE LEGAL PROSE

The legal prose of the contract

This contract represents an agreement between **ACME Pte Ltd** and **EMCA Pte Ltd**.

Mauris porttitor varius diam vel pellentesque. Etiam id **10** velit. Curabitur ac congue arcu. Aliquam lectus ligula, ullamcorper et ullamcorper eu, venenatis ut quam.

Curabitur quis orci tempor, commodo libero sit amet, sodales velit. Interdum et malesuada fames ac ante ipsum primis in faucibus. Aenean ex sem, **5%** after **1 month** leo non, dictum consectetur quam. Aliquam eu porta nibh.

Pellentesque lacinia, ex eu imperdiet pretium, metus sem commodo elit, at luctus mauris sed lectus. Phasellus imperdiet non orci semper consectetur. Vestibulum a purus tellus. Praesent ornare imperdiet suscipit. In eu consectetur nibh.

The grammar of the contract

This contract represents an agreement between **[{seller}]** and **[{buyer}].**

Mauris porttitor varius diam vel pellentesque. Etiam id **[{numberOfItems}]** velit. Curabitur ac congue arcu. Aliquam lectus ligula, ullamcorper et ullamcorper eu, venenatis ut quam.

Curabitur quis orci tempor, commodo libero sit amet, sodales velit. Interdum et malesuada fames ac ante ipsum primis in faucibus. Aenean ex sem, **[{penaltyPercetage}]** **[{penaltyDuration}]** leo non, dictum consectetur quam. Aliquam eu porta nibh.

Pellentesque lacinia, ex eu imperdiet pretium, metus sem commodo elit, at luctus mauris sed lectus. Phasellus imperdiet non orci semper consectetur. Vestibulum a purus tellus. Praesent ornare imperdiet suscipit. In eu consectetur nibh.

Data model

```
asset DeliveryAndPenaltyContract extends Contract {
        --> Party seller
        --> Party buyer
        o Integer numberOfItems
        o Double penaltyPercetage
        o Duration penaltyDuration
}
```

Contract logic

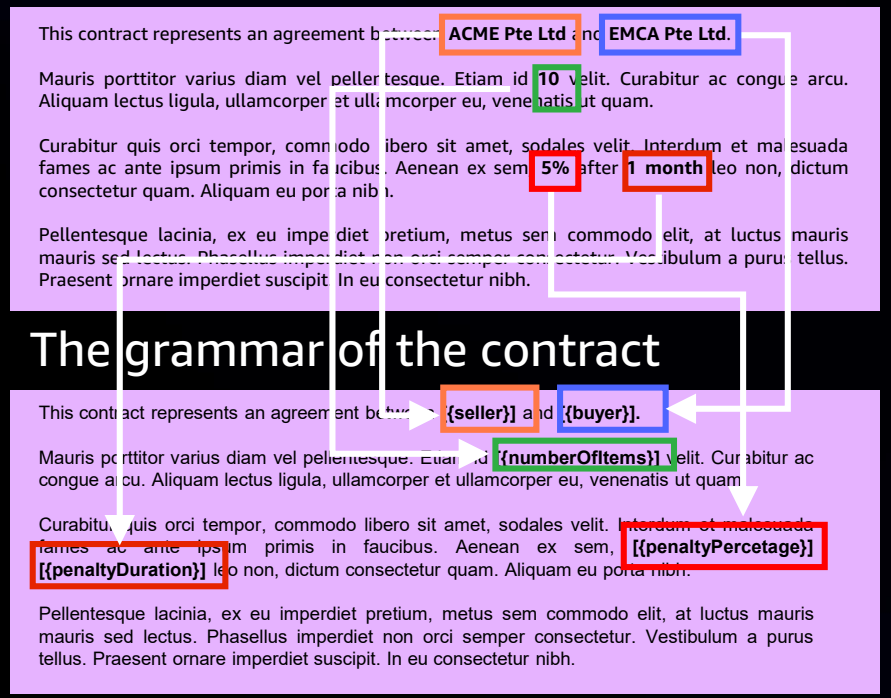
```
contract DeliveryAndPenalty over DeliveryAndPenaltyContract {
  clause latedeliveryandpenalty(request : LateRequest) : LateResponse emits PaymentObligation {
    let agreed = request.agreedDelivery;
    let delivery = request.deliveredAt;
    enforce isAfter(delivery, agreed)
    else throw Error{ message : "Cannot exercise late delivery before delivery date" };
    // 1. Penalty formula
    let delay = diffDurationAs(delivery,agreed,~org.accordproject.time.TemporalUnit.days);      // Delay in days
    let penaltyStep = contract.penaltyDuration;              // Penalty steps
    // 2. Emit a payment obligation in USD
    emit PaymentObligation{
      contract: contract,
      promisor: some(contract.seller),
      promisee: some(contract.buyer),
      deadline: none,
      amount: MonetaryAmount{ doubleValue: penalty, currencyCode: USD },
      description: toString(contract.seller) ++ " should pay penalty amount to " ++ toString(contract.buyer)
    };
    // 3. Return the response
    return LateResponse{
      penalty: penalty
    }
  }
}
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Introducing the Accord Project

2. EXTRACT CONTRACT DATA FROM THE LEGAL PROSE WITH GRAMMAR

The legal prose of the contract



The grammar of the contract

Data model

```
asset DeliveryAndPenaltyContract extends Contract {
        --> Party seller
        --> Party buyer
        o Integer numberOfItems
        o Double penaltyPercetage
        o Duration penaltyDuration
}
```

Contract logic

```
contract DeliveryAndPenalty over DeliveryAndPenaltyContract {
 clause latedeliveryandpenalty(request : LateRequest) : LateResponse emits PaymentObligation {
  let agreed = request.agreedDelivery;
  let delivery = request.deliveredAt;
  enforce isAfter(delivery, agreed)
  else throw Error{ message : "Cannot exercise late delivery before delivery date" };
  // 1. Penalty formula
  let delay = diffDurationAs(delivery,agreed,~org.accordproject.time.TemporalUnit.days);      // Delay in days
  let penaltyStep = contract.penaltyDuration;             // Penalty steps
  // 2. Emit a payment obligation in USD
  emit PaymentObligation{
   contract: contract,
   promisor: some(contract.seller),
   promisee: some(contract.buyer),
   deadline: none,
   amount: MonetaryAmount{ doubleValue: penalty, currencyCode: USD },
   description: toString(contract.seller) ++ " should pay penalty amount to " ++ toString(contract.buyer)
  };
  // 3. Return the response
  return LateResponse{
   penalty: penalty
  }
 }
}
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Introducing the Accord Project

3. MAP GRAMMAR TO THE DATA MODEL

The legal prose of the contract

This contract represents an agreement between **ACME Pte Ltd** and **EMCA Pte Ltd**.

Mauris porttitor varius diam vel pellentesque. Etiam id **10** velit. Curabitur ac congue arcu. Aliquam lectus ligula, ullamcorper et ullamcorper eu, venenatis ut quam.

Curabitur quis orci tempor, commodo libero sit amet, sodales velit. Interdum et malesuada fames ac ante ipsum primis in faucibus. Aenean ex sem, **5%** after **1 month** leo non, dictum consectetur quam. Aliquam eu porta nibh.

Pellentesque lacinia, ex eu imperdiet pretium, metus sem commodo elit, at luctus mauris sed lectus. Phasellus imperdiet non orci semper consectetur. Vestibulum a purus tellus. Praesent ornare imperdiet suscipit. In eu consectetur nibh.

The grammar of the contract

This contract represents an agreement between **{seller}** and **{buyer}**.

Mauris porttitor varius diam vel pellentesque. Etiam id **{numberOfItems}** velit. Curabitur ac congue arcu. Aliquam lectus ligula, ullamcorper et ullamcorper eu, venenatis ut quam.

Curabitur quis orci tempor, commodo libero sit amet, sodales velit. Interdum et malesuada fames ac ante ipsum primis in faucibus. Aenean ex sem, **[{penaltyPercetage}]** **[{penaltyDuration}]** leo non, dictum consectetur quam. Aliquam eu porta nibh.

Pellentesque lacinia, ex eu imperdiet pretium, metus sem commodo elit, at luctus mauris sed lectus. Phasellus imperdiet non orci semper consectetur. Vestibulum a purus tellus. Praesent ornare imperdiet suscipit. In eu consectetur nibh.

Data model

```
asset DeliveryAndPenaltyContract extends Contract {
    --> Party seller
    --> Party buyer
    o Integer numberOfItems
    o Double penaltyPercetage
    o Duration penaltyDuration
}
```

Contract logic

```
contract DeliveryAndPenalty over DeliveryAndPenaltyContract {
  clause latedeliveryandpenalty(request : LateRequest) : LateResponse emits PaymentObligation {
    let agreed = request.agreedDelivery;
    let delivery = request.deliveredAt;
    enforce isAfter(delivery, agreed)
    else throw Error{ message : "Cannot exercise late delivery before delivery date" };
    // 1. Penalty formula
    let delay = diffDurationAs(delivery,agreed,~org.accordproject.time.TemporalUnit.days);      // Delay in days
    let penaltyStep = contract.penaltyDuration;              // Penalty steps
    // 2. Emit a payment obligation in USD
    emit PaymentObligation{
      contract: contract,
      promisor: some(contract.seller),
      promisee: some(contract.buyer),
      deadline: none,
      amount: MonetaryAmount{ doubleValue: penalty, currencyCode: USD },
      description: toString(contract.seller) ++ " should pay penalty amount to " ++ toString(contract.buyer)
    };
    // 3. Return the response
    return LateResponse{
      penalty: penalty
    }
  }
}
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Introducing the Accord Project

4. RUN LOGIC WITH CONTRACT DATA

The legal prose of the contract

This contract represents an agreement between **ACME Pte Ltd** and **EMCA Pte Ltd**.

Mauris porttitor varius diam vel pellentesque. Etiam id **10** velit. Curabitur ac congue arcu. Aliquam lectus ligula, ullamcorper et ullamcorper eu, venenatis ut quam.

Curabitur quis orci tempor, commodo libero sit amet, sodales velit. Interdum et malesuada fames ac ante ipsum primis in faucibus. Aenean ex sem, **5%** after **1 month** leo non, dictum consectetur quam. Aliquam eu porta nibh.

Pellentesque lacinia, ex eu imperdiet pretium, metus sem commodo elit, at luctus mauris mauris sed lectus. Phasellus imperdiet non orci semper consectetur. Vestibulum a purus tellus. Praesent ornare imperdiet suscipit. In eu consectetur nibh.

The grammar of the contract

This contract represents an agreement between **{seller}** and **{buyer}**.

Mauris porttitor varius diam vel pellentesque. Etiam id **{numberOfItems}** velit. Curabitur ac congue arcu. Aliquam lectus ligula, ullamcorper et ullamcorper eu, venenatis ut quam.

Curabitur quis orci tempor, commodo libero sit amet, sodales velit. Interdum et malesuada fames ac ante ipsum primis in faucibus. Aenean ex sem, **[{penaltyPercetage}]** **[{penaltyDuration}]** leo non, dictum consectetur quam. Aliquam eu porta nibh.

Pellentesque lacinia, ex eu imperdiet pretium, metus sem commodo elit, at luctus mauris mauris sed lectus. Phasellus imperdiet non orci semper consectetur. Vestibulum a purus tellus. Praesent ornare imperdiet suscipit. In eu consectetur nibh.

Data model

```
asset DeliveryAndPenaltyContract extends Contract {
        --> Party seller
        --> Party buyer
        o Integer numberOfItems
        o Double penaltyPercetage
        o Duration penaltyDuration
}
```

Contract logic

```
contract DeliveryAndPenalty over DeliveryAndPenaltyContract {
  clause latedeliveryandpenalty(request : LateRequest) : LateResponse emits PaymentObligation {
    let agreed = request.agreedDelivery;
    let delivery = request.deliveredAt;
    enforce isAfter(delivery, agreed)
    else throw Error{ message : "Cannot exercise late delivery before delivery date" };
    // 1. Penalty formula
    let delay = diffDurationAs(delivery,agreed,~org.accordproject.time.TemporalUnit.days);      // Delay in days
    let penaltyStep = contract.penaltyDuration;          // Penalty steps
    // 2. Emit a payment obligation in USD
    emit PaymentObligation{
      contract: contract,
      promisor: some(contract.seller),
      promisee: some(contract.buyer),
      deadline: none,
      amount: MonetaryAmount{ doubleValue: penalty, currencyCode: USD },
      description: toString(contract.seller) ++ " should pay penalty amount to " ++ toString(contract.buyer)
    };
    // 3. Return the response
    return LateResponse{
      penalty: penalty
    }
  }
}
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Introducing the Accord Project

5. THE RESULT IS A SINGLE JSON DOCUMENT

The unique ID and hash of the template

Request details by a user

Response details by the logic

New contract state

Array of events emitted


```
{
"clause": "minilatedeliveryandpenalty-payment@0.5.0-
6ba8b3df91608c316b8bdffde2131f2816e5dfdcc5070ec007ba274ed730e2c9",
"request": {
        "$class": "org.accordproject.minilatedeliveryandpenalty.LateRequest",
        "agreedDelivery": "2019-04-01T12:00:00-05:00",
        "deliveredAt": "2019-04-20T03:24:00-05:00",
        "goodsValue": 200
    },
"response": {
        "$class": "org.accordproject.minilatedeliveryandpenalty.LateResponse",
        "penalty": 104,
        "buyerMayTerminate": true,
        "transactionId": "9306d752-1457-46c8-888d-70c824ec1094",
        "timestamp": "2020-10-12T07:51:15.840Z"
    },
"state": {
        "$class": "org.accordproject.cicero.contract.AccordContractState",
        "stateId": "org.accordproject.cicero.contract.AccordContractState#1"
    },
"emit": [{
        "$class": "org.accordproject.cicero.runtime.PaymentObligation",
        "amount": {
            "$class": "org.accordproject.money.MonetaryAmount",
            "doubleValue": 104,
            "currencyCode": "USD"
            },
        "description": "Betty Buyer should pay penalty amount to Steve Seller",
        "contract":
        "resource:org.accordproject.minilatedeliveryandpenalty.MiniLateDeliveryContract#df1
        2432a-0a82-47d0-9dcb-f3f4527cba32",
        "promisor": "resource:org.accordproject.cicero.contract.AccordParty#Betty%20Buyer",
        "promisee": "resource:org.accordproject.cicero.contract.AccordParty#Steve%20Seller",
        "eventId": "adljaADFAjwrayy234",
        "timestamp": "2020-10-12T07:51:15.842Z"
    }]
}
```


© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon QLDB characteristics

FULLY MANAGED LEDGER DATABASE SERVICE

Immutable



Append-only, immutable journal tracks history of all changes

Built-in data lineage, available to users who were granted access by the ledger owner

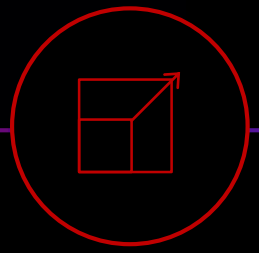
Verifiable



All changes are cryptographically chained and verifiable

Verification can be performed through an API or by exporting the ledger blocks to Amazon S3

Serverless



Serverless, no need to provision storage or IO in advance

Scalable and highly available

Easy to use

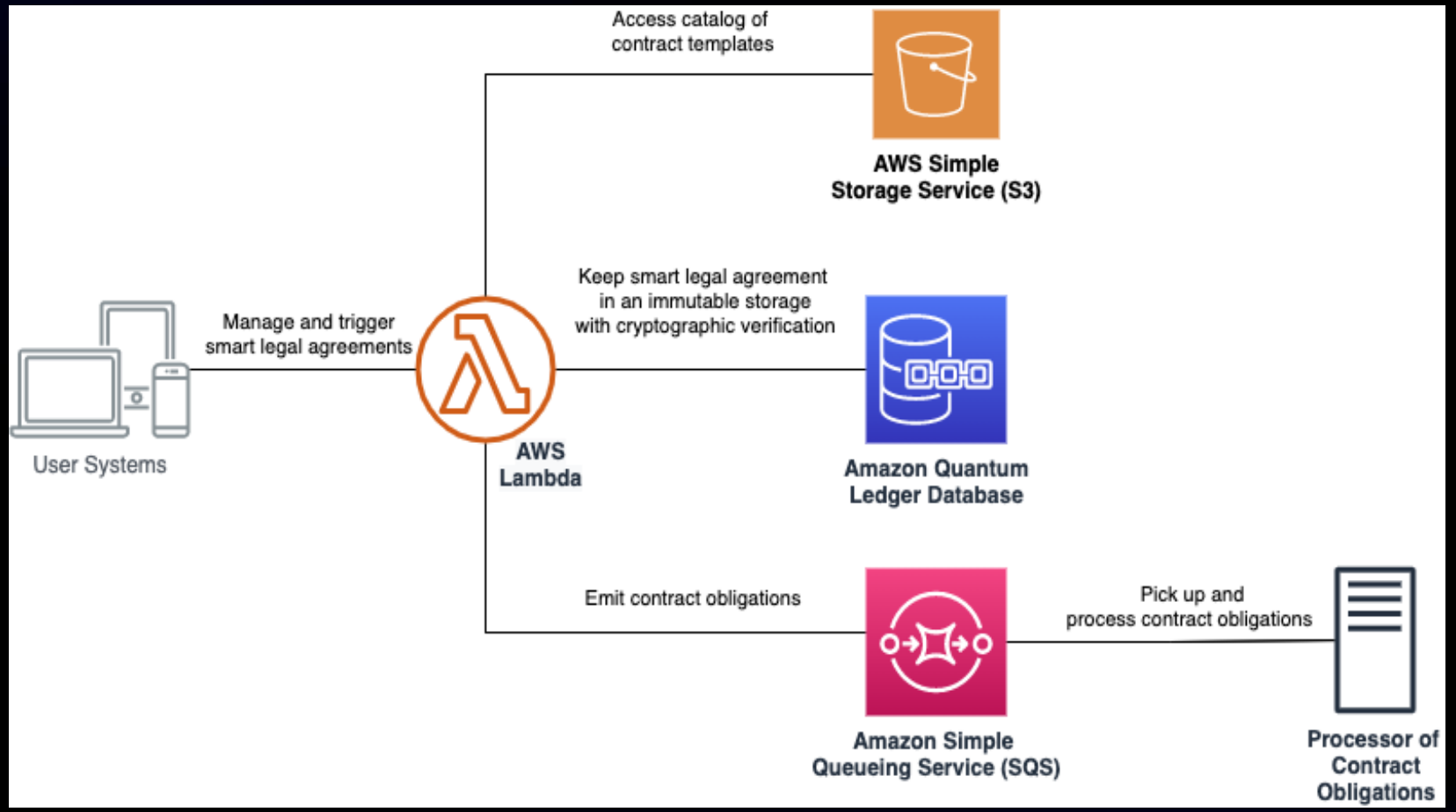


PartiQL offers familiar SQL operators

ION data model is rich and expressive

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

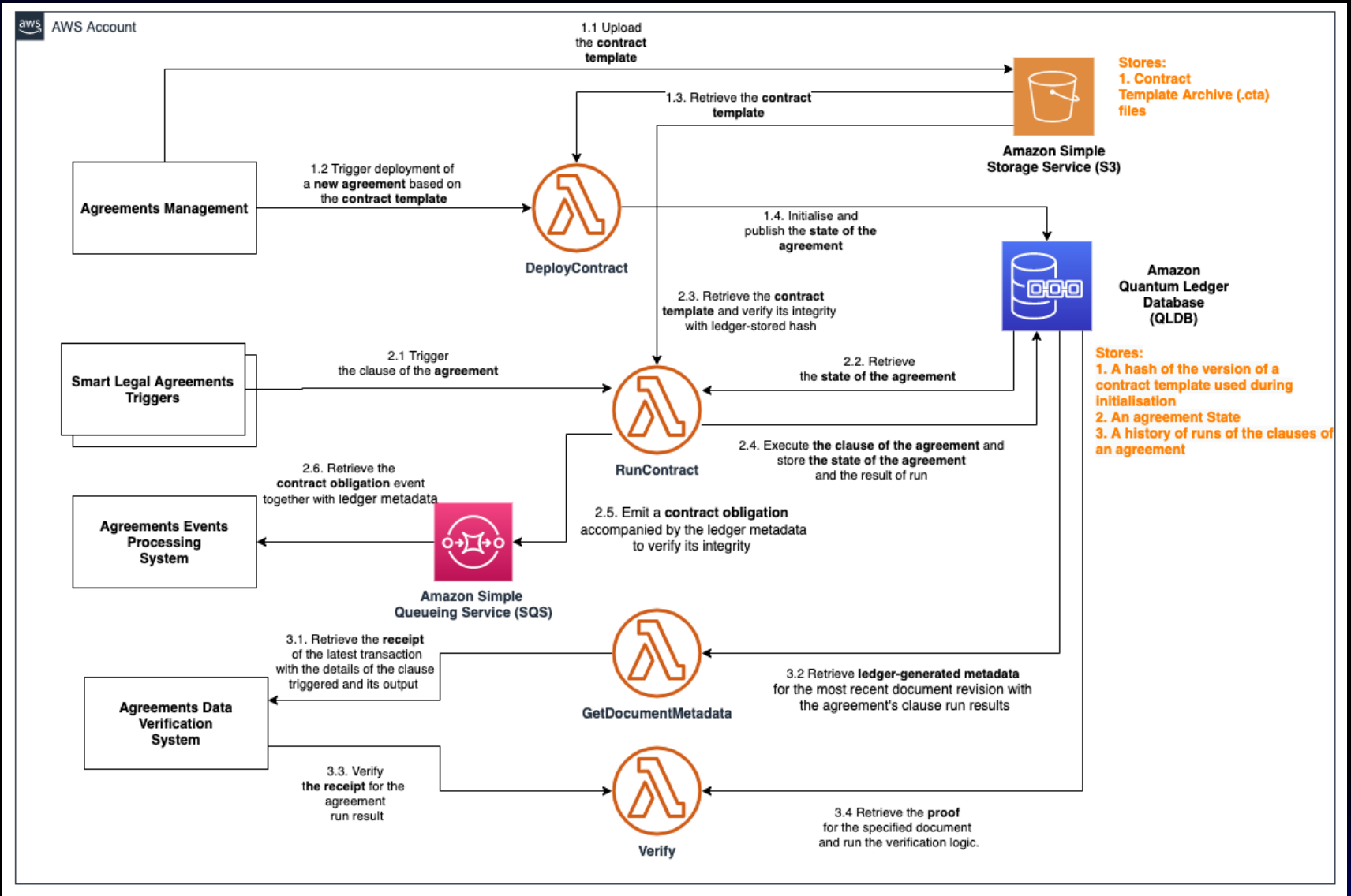
Running the Accord Project on Amazon QLDB



https://go.aws/2YkCxlk
https://go.aws/3bLlJad

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Smart legal contract flow



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Rafia Tapia
taprafia@amazon.com

Dan Blaner
dbblaner@amazon.com

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.