# Adding Remote Database and SSL Support

**Nick Russo**

NETWORK ENGINEER

@nickrusso42518   www.njrusmc.net

# Agenda

**Understanding different database types**

**Object-relational Mappers (ORM)**

**Adding MySQL support to the CRM app**

**Improving security with SSL**

# How Do Relational Databases Work?

table "account"

| acctid (string) | paid (float) | due (float) |
|---|---|---|
| ACCT100 | 60.00 | 100.00 |
| ACCT200 | 70.00 | 60.00 |
| ACCT300 | 0.00 | 0.00 |

*primary key*

"I need to compute the balance for ACCT200"

# ACID for Databases
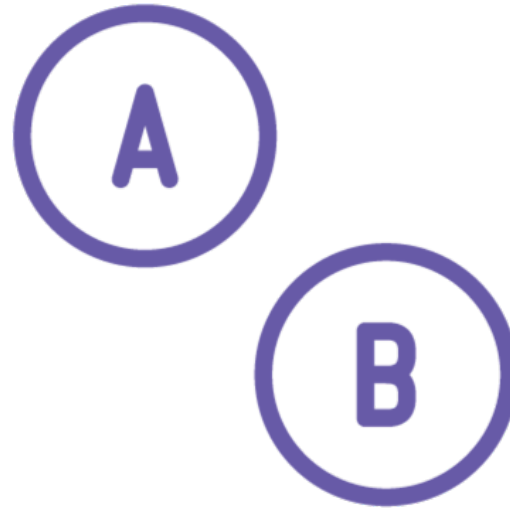
**Atomic**　　　**Consistent**　　　**Isolation**　　　**Durable**

```json
[
    {
        "acctid": "ACCT100",
        "paid": 60.00,
        "due": 100.00,
        "name: "John Smith"
    },
    {
        "acctid": "ACCT200",
        "paid": 70.00,
        "due": 60.00,
    },
    {
        "acctid": "ACCT300",
        "paid": 0.00,
        "due": 0.00,
    }
]
```

◄ Each dictionary is a "document"

◄ Collection of dictionaries is a "book"
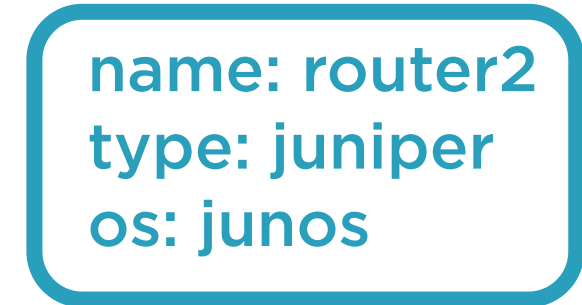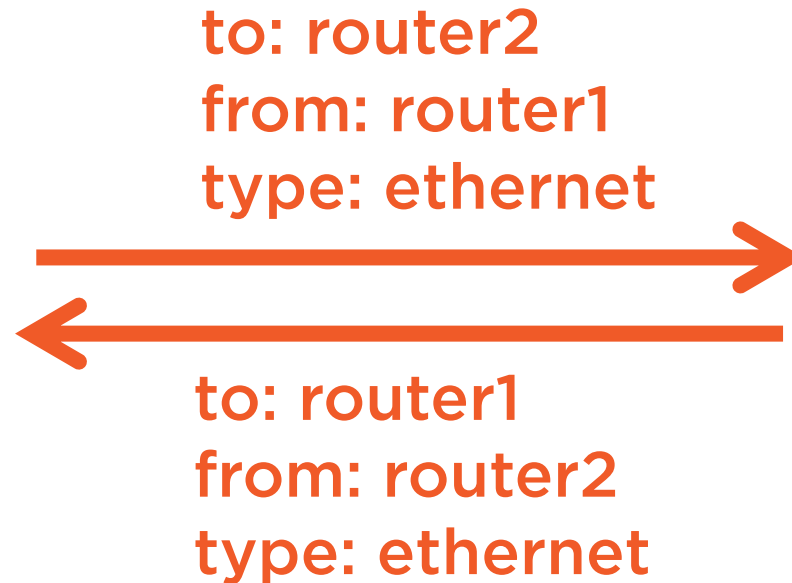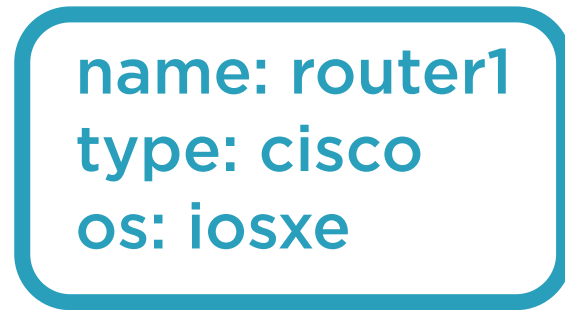
◄ Can add extra, optional fields easily

eCommerce

Content
(blog/video)

# Non-relational Databases: Graph

**Nodes in blue**
**Edges in orange**

name: router1
type: cisco
os: iosxe

to: router2
from: router1
type: ethernet

to: router1
from: router2
type: ethernet

name: router2
type: juniper
os: junos

"I want to traverse from router1 to router2 to ..."

# Alternative Databases: Columnar

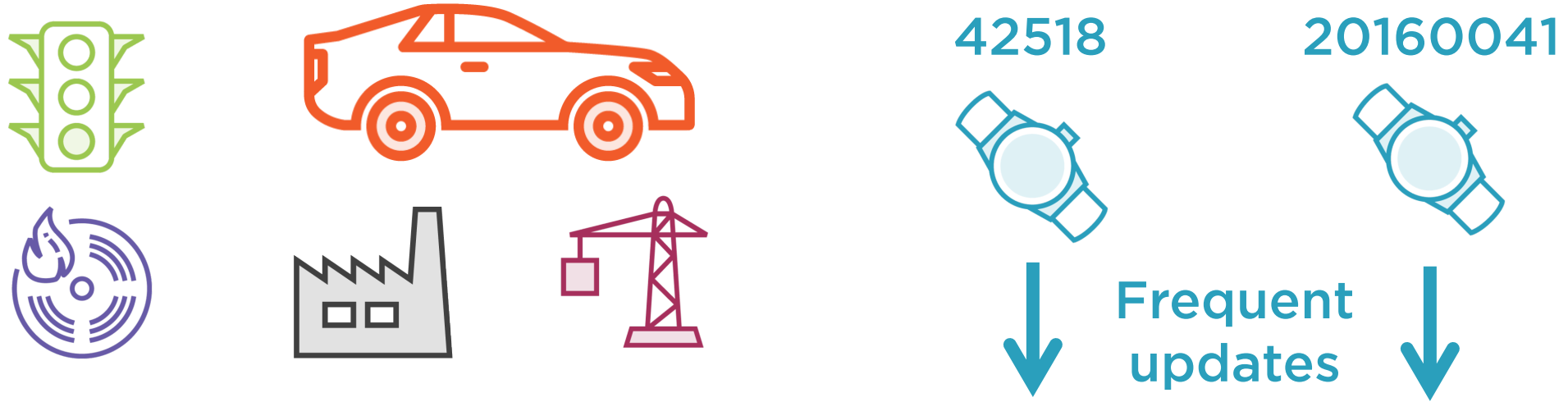| acctid (string) | paid (float) | due (float) |
|---|---|---|
| ACCT100 | 60.00 | 100.00 |
| ACCT200 | 70.00 | 60.00 |
| ACCT300 | 0.00 | 0.00 |

"How much are we owed, on average?"

**Row-based**

ACCT100,60.00,100.00;ACCT200,70.00,60.00;...

**Column-based**

ACCT100,ACCT200,ACCT300;60.00,70.00,0.00;...

# Non-relational Databases: Time Series

42518          20160041

Frequent updates

| timestamp | metric_id | sensor_id | value |
|---|---|---|---|
| 2019-10-07T00:00:00Z | temp | 42518 | 30 |
| 2019-10-07T00:00:00Z | battery | 42518 | 84 |
| 2019-10-07T00:00:03Z | temp | 20160041 | 28 |
| 2019-10-07T00:00:03Z | battery | 20160041 | 97 |

# Database Selection at a Glance

**Relational:** fixed field structure based on schema, easy to understand

**Document:** good for arbitrarily structured or deeply nested data

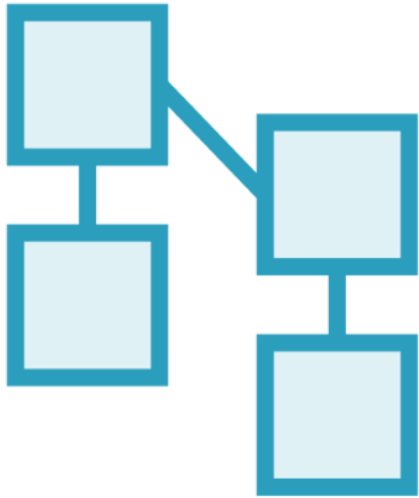**Graph:** data set defined by relationships supporting node traversal

**Columnar:** data aggregation (average, total, etc.) from many records

**Time series:** sensor data/telemetry with many entries per reporter

# Object-relational Mappers (ORM)

**Abstraction**

**Easy and safe**

# SQL ORM: SQLAlchemy



```
def make_table(db_url):
    engine = create_engine(db_url)
    meta = MetaData(engine)
    table = Table(<new table args>)
```

**"I need to ..."**

**"... start small."**

```
db_url =
"mysql://root:pass123
@host/db"
```

**"... introduce new features."**

```
db_url =
"postgres://admin:pass
123@host/db"
```

**"... build tests."**

```
db_url =
"sqlite:///:memory:"
```

# Demo

## Using MySQL with Python SQLAlchemy

# Demo

**Does the app still work?**

# What Is SSL?



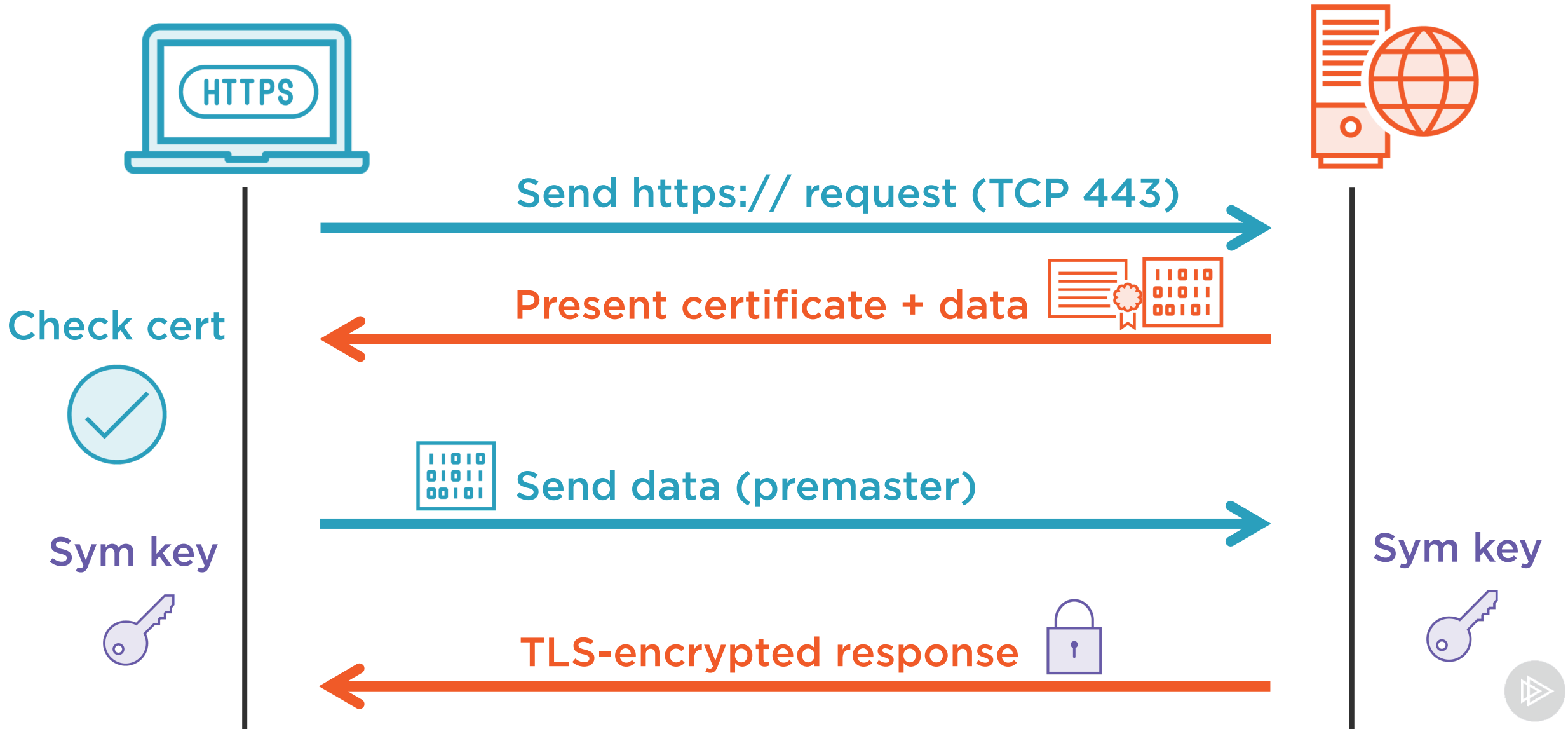**Digital Certificates**

**Data Confidentiality and Integrity**

**End-to-end encryption**

**Also known as Transport Layer Security (TLS)**

# High-level TLS Operation in HTTPS

**HTTPS**

Send https:// request (TCP 443)

Present certificate + data

**Check cert**

Send data (premaster)

**Sym key**

**Sym key**

TLS-encrypted response

# Demo

**Adding self-signed SSL certificates to the CRM app**

# Summary

**Many types of databases**

**Many languages have ORMs (use them!)**

**SSL/TLS ... tricky but necessary**