

IBM Blockchain Hands-On

IBM Blockchain Platform:

Learn Blockchain Concepts

Lab One

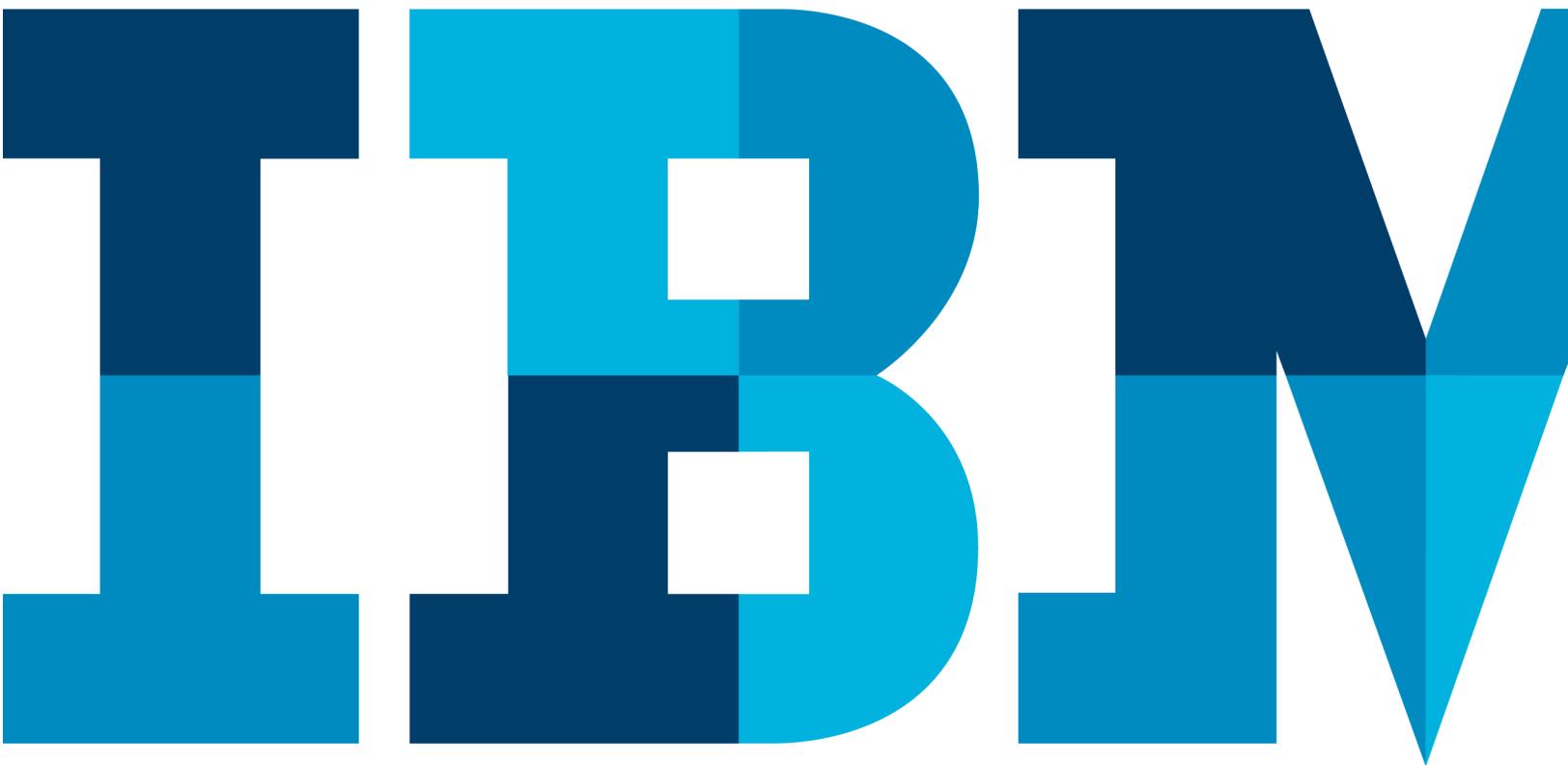


Table of Contents

- Disclaimer..... 3**
- 1 Overview of the lab 1 environment and scenario 5**
- 2 Running the lab..... 7**
 - 2.1 Ordering the car 7
 - 2.2 Manufacturing the car12
 - 2.3 Insuring the car15
- 3 Cool Stuff.....20**
 - 3.1 Internet of Things Integration 21
 - 3.2 Analytics25
- 4 Next Steps27**
- Appendix A. Restarting the sample application.....29**
- Appendix B. Changing the keyboard language.....30**

Disclaimer

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.

The development, release, and timing of any future features or functionality described for our products remains at our sole discretion I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results like those stated here.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **This document is distributed "as is" without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.** IBM products and services are warranted per the terms and conditions of the agreements under which they are provided.

IBM products are manufactured from new parts or new and used parts. In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and

discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products about this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, ibm.com and [names of other referenced IBM products and services used in the presentation] are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.

© 2019 International Business Machines Corporation. No part of this document may be reproduced or transmitted in any form without written permission from IBM.

U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

1 Overview of the lab 1 environment and scenario

This lab allows you to experience a blockchain solution from the point of view of a set of end-users, and in doing so learn about key blockchain concepts. It is not meant to be a technical introduction to blockchain, but will instead focus on the properties of the business network and value that blockchain brings.

The use-case we will work through is one that demonstrates the **lifecycle of a new car**, from the manufacturing and purchasing through to delivery and insurance. It is a good blockchain use-case because there is a defined business network and an identifiable need for trust between the participants of the network.

In this lab, you will be playing the role of the four personas who use the vehicle lifecycle system:

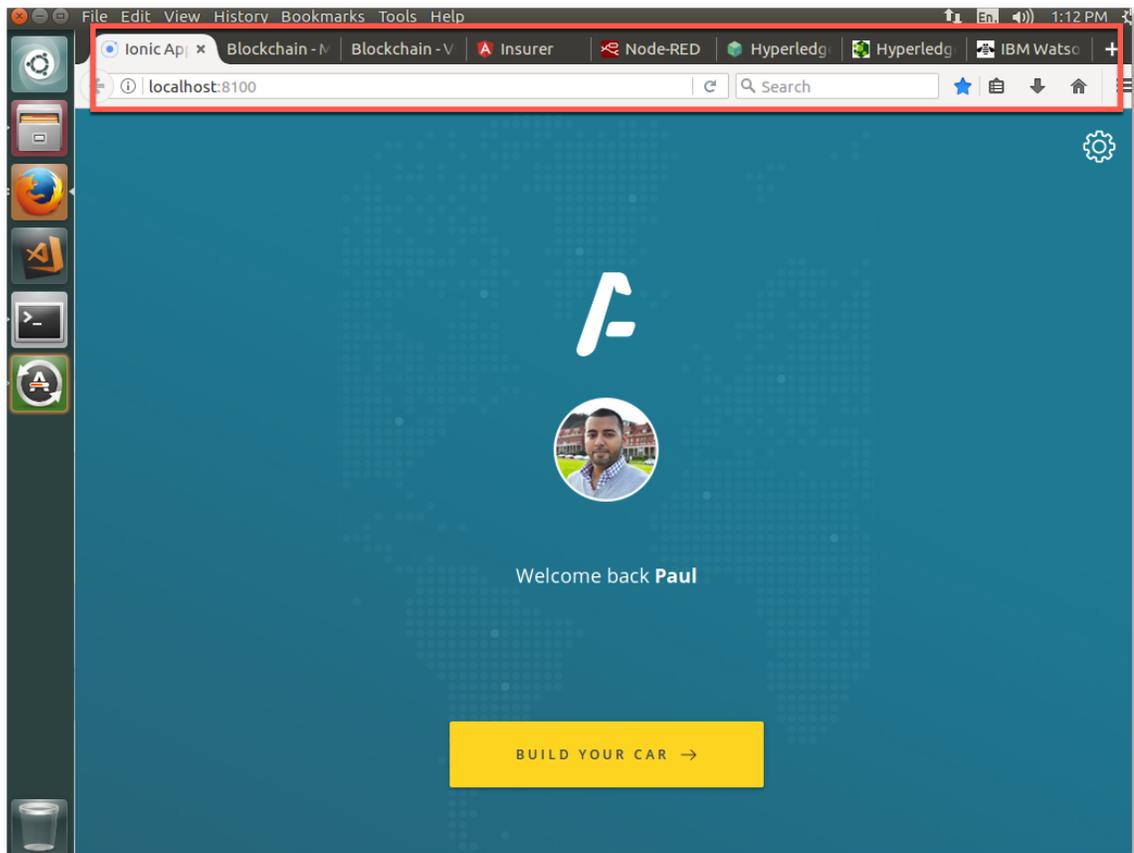
- *Paul* - the buyer/owner of a car
- *Mike* - an employee for the car manufacturer (“Arium”)
- *Debbie* - an administrator for the regulator (“Vehicle & Drivers Authority” or “VDA”)
- *Tommen* - an Insurer from an insurance company called Prince

These personas together work on ordering, building, transferring ownership of a vehicle while keeping all the other parties in the network updated and building the trust between them to allow them to work together efficiently.

In this lab, each user’s application will be represented by a separate tab in our web browser; of course, in a real blockchain network they each user will be running on different systems in different locations, although all connecting in to the same (but distributed) blockchain network.

Start here. Instructions are always shown on numbered lines like this one:

- 
- __ 1. If it is not already running, start the virtual machine for the lab. The instructor will tell you how to do this if you are unsure.
 - __ 2. Wait for the image to boot, and for the blockchain application and associated services to start. This happens automatically but might take several minutes. The image is ready to use when the web browser is visible and eight tabs fully loaded, as per the screenshot below.



While the virtual machine is starting, let's recap a few blockchain concepts and introduce the scenario.

The most generally accepted definition of a blockchain is of a **shared, replicated ledger**.

Ledgers have been around for hundreds of years and are records of what a business has done. They're important systems of record because they describe a business's inputs and outputs and thereby give an indication of its worth. Essentially, they are a log of transactions – a transaction being a change in state of an asset.

The problem with ledgers is that each one is owned by a single business, which means that when one business transacts with another, ledgers can get out of sync. What happens when the transaction I've recorded on my ledger doesn't tally with the transaction you've recorded on your ledger? Disputes inevitably occur which need to be resolved through a reconciliation process. This can be slow and expensive.

By having a *shared* ledger it means that all participants of the business network see the same ledger. By *replicating* it across the business network, it means that the ledger is not held in any one single place, which would otherwise make it vulnerable to outages and malicious attack.

Consider the business network that surrounds the purchase and ownership of a car. Today, each participant (for example, manufacturer, insurer or regulator) has their own ledger and the processes that allow them to interact with each other varies from company to company, and can be time consuming to complete. Connectivity between participants is typically done point-to-point using a variety of processes – some manual or slow (e.g. sending a letter), and some automated (e.g. file, REST API, B2B messaging). This plethora of processes is expensive to maintain and can be vulnerable to attack.

In this scenario we will replace these disparate ledgers with a single blockchain, and the individual business processes with a single shared one. By doing this we will make the overall process much quicker and less prone to error.

We will experience the solution through the eyes of four key members of the business network: a private purchaser, the manufacturer, the regulator and the insurer.

We will start by looking at the ordering process, as experienced by the buyer.

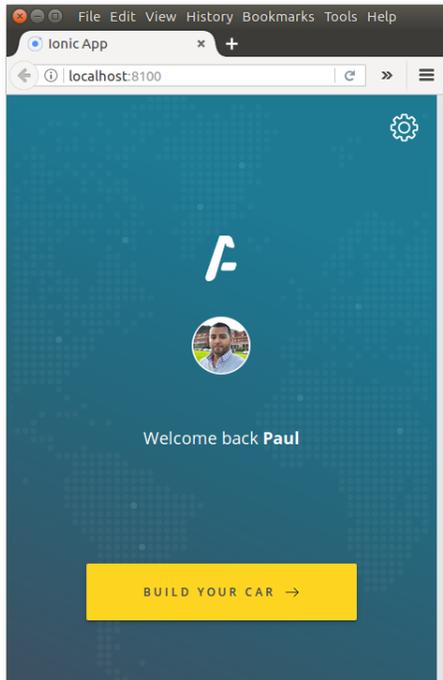
2 Running the lab

2.1 Ordering the car

3. If it is not already selected, Switch to the “Ionic App” tab on the web browser (running at localhost:8100).

Note that Paul’s web page is intended to be viewed as a mobile app; you might want to ungroup this tab from the others by dragging the tab’s title bar off from the others, and resizing the window to make it easier to view and navigate as a mobile app.

IBM Blockchain

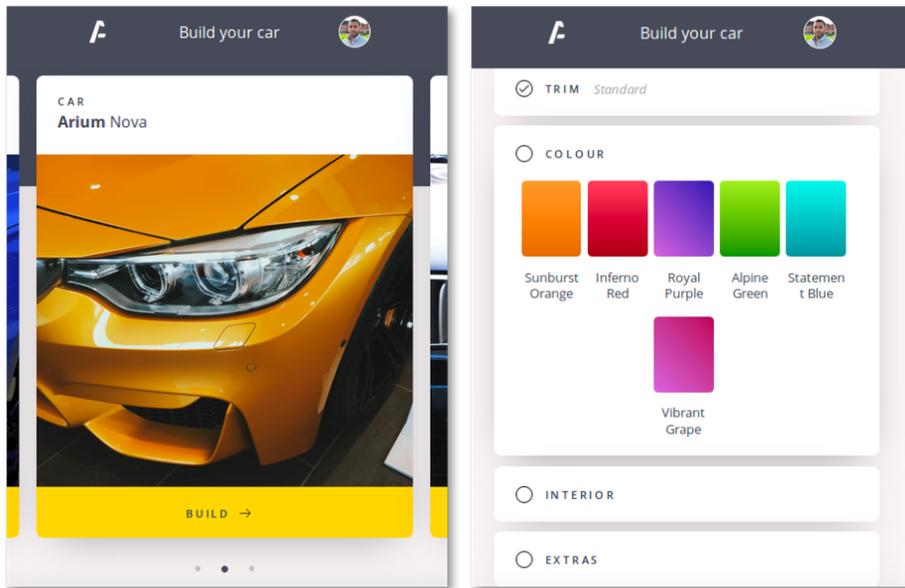


__ 4. In Paul's app, click **'Build Your Car'**.

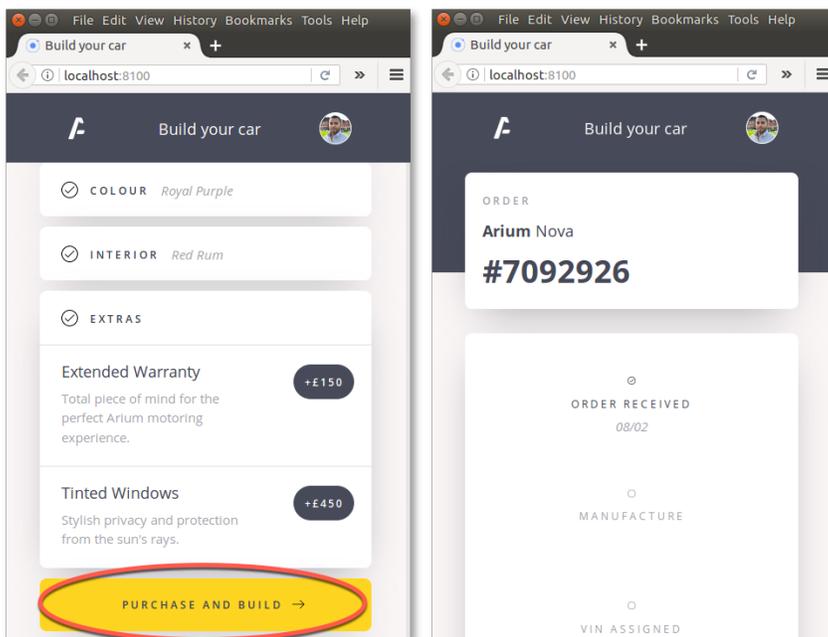


IBM Blockchain

___ 5. Swipe left and right to decide which car to build, and then decide the options on Paul's car.



___ 6. Once you have decided on each of Paul's options, click 'Purchase and Build'.



___ 7. Once you place the order, switch to the “Blockchain – VDA” tab (localhost:6001/dashboard).

As you will recall, the VDA is the regulator who requires notification of all transactions that occur within the business network. Debbie, who works for the regulator, has a dashboard running on her PC that shows all transactions as they occur.

You will immediately see the VDA dashboard update itself with the latest transaction.

The screenshot shows a web browser window with multiple tabs. The active tab is 'Blockchain - VDA' at the URL 'localhost:6001/dashboard'. The dashboard header includes the 'Vehicle & Drivers Authority' logo and a search bar. Below the header are four summary cards: 'REGISTERED VEHICLES' (16), 'VIN ASSIGNED' (16), 'ASSET ACTIVITY' (22), and 'SUSPICIOUS VEHICLES' (5). A dropdown menu shows 'IN THE LAST WEEK'. The main content area displays a horizontal list of transactions from #154 to #160, each with a transaction ID and a description. Transaction #160 is highlighted as the 'NEW TRANSACTION' and is 'UpdateOrderStatus'. Below this is a 'RECENT TRANSACTIONS' table with columns for Timestamp, Transaction ID, Transaction Type, and Transaction Submitter. The table lists two recent transactions: an 'UpdateOrderStatus' transaction from 'Arium Vehicles' and a 'PlaceOrder' transaction from 'Paul Harris'.

| Timestamp | Transaction ID | Transaction Type | Transaction Submitter |
|------------------------|--|-------------------|-----------------------|
| Feb 8, 2018 2:43:20 PM | a6ad00339e57d8dd6e9f70131e87ededc6c17ee0c3415155eb7e9b1f835beebc | UpdateOrderStatus | Arium Vehicles |
| Feb 8, 2018 2:43:17 PM | e11f4bb6bac7ebee2f5074d658d40d7e25608dbcce9aecc2d3714f28a244cd3b | PlaceOrder | Paul Harris |

If you look at the “Recent Transactions” log at the bottom of the page, you will see two new transactions listed: a “PlaceOrder” transaction submitted by Paul Harris, and an “UpdateOrderStatus” acknowledgement from Arium Vehicles, our manufacturer.

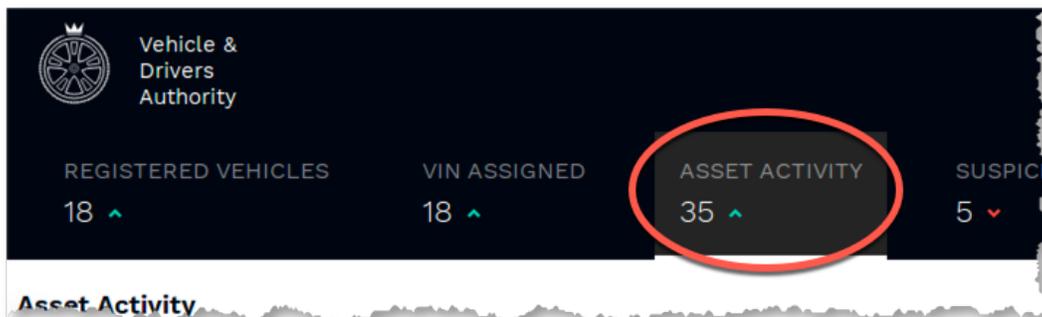
| | | | |
|---------------------------|--|-------------------|----------------|
| Feb 8, 2018 2:43:20 PM | a6ad00339e57d8dd6e9f70131e87ededc6c17ee0c3415155eb7e9b1f835beebc | UpdateOrderStatus | Arium Vehicles |
| Feb 8, 2018 2:43:17 PM | e11f4bb6bac7ebee2f5074d658d40d7e25608dbcce9aecc2d3714f28a244cd3b | PlaceOrder | Paul Harris |

Look in the blue section above this log and you will see those transactions represented graphically as a chain, with the most recent transactions on the right. This is a representation of our blockchain, and the regulator can see everything that is stored on it.



As you will recall, the blockchain is our transaction log which is shared (with appropriate privacy and permissioning) between the participants of our business network. Each block in this chain could potentially actually contain multiple transactions, but here you'll just see each unique transaction inside its own block.

___ 8. Click on 'Asset activity' within the VDA dashboard.



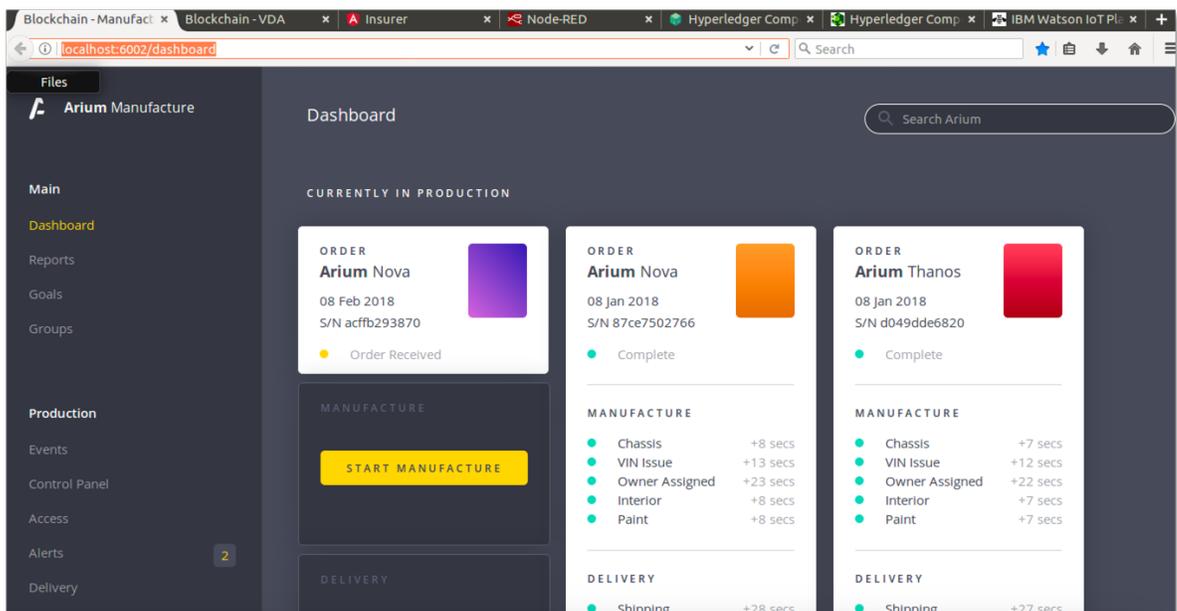
This is an alternative view of the ledger that shows all the transactions that have occurred, and the participants involved.

| Asset Activity | | Transaction Validators | |
|--|--|------------------------|------------------------|
| Transaction Type | | | |
| New Insurance Issued F431500378a210add59dcda4b09761d47da59b96877401ccf0d7d53465ec06e3 | | <u>Insurer</u> | + <u>Vehicle Owner</u> |
| Vehicle Manufacture (Delivered) D32491bf58d15bb138d6e15991a87eb67ed4841a94199bcb3d0782e39614506c | | <u>Vehicle Owner</u> | + <u>Vehicle</u> |
| Vehicle Manufacture (Owner Assigned) 86c68961764ce6acea2c9adb23376241369f1376650aa4bff078fd0e90f95d1 | | <u>Manufacturer</u> | |
| New Usage Record 3d64010b278077097b9bd02c924dc5403a7e3e8e33b1aa5a4d174d5899b451f6 | | <u>Manufacturer</u> | |

2.2 Manufacturing the car

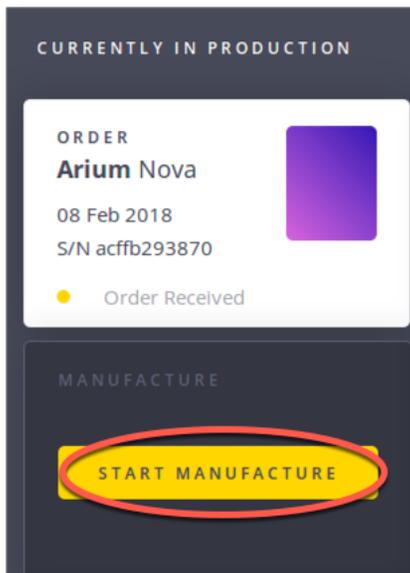
9. Switch to the “Blockchain – Manufacturer” tab (localhost:6002/dashboard).

This is the dashboard that Mike, who works for Arium, uses. He does not have full visibility into the entire blockchain that the regulator requires, but can see the parts of it that pertain to Arium: specifically, he has visibility into all the orders that are coming in so that he can control the manufacturing process.



The “Currently in Production” section of this page shows those orders that have been received and the cars that have recently been built. The left-most order in this section will be the car that Paul recently ordered.

- __ 10. Click Start Manufacture underneath Paul’s order to start the business process to build a car.



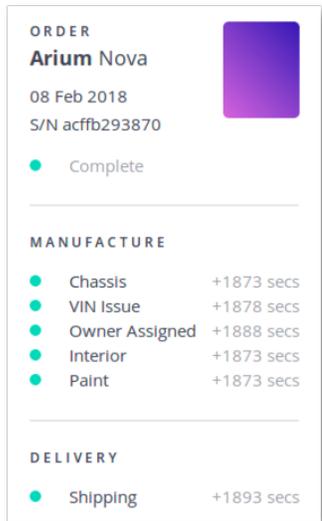
The production process has (of course) been simulated and will take place over the next several seconds; the vehicle will be ‘built’ and blockchain transactions submitted that record status at key milestones of the production process. In a real network, different automated plant systems will trigger these events, which are signed off by the manufacturer, and the issuance of the Vehicle Identification Number might be signed off by the regulator.

- __ 11. As the car is being built, switch back to the VDA dashboard to see these key milestones being represented on the unfiltered blockchain.

IBM Blockchain



- ___ 12. Also note how the Manufacturer’s view changes as the vehicle is being built, with icons changing to green as those parts of the process are completed.

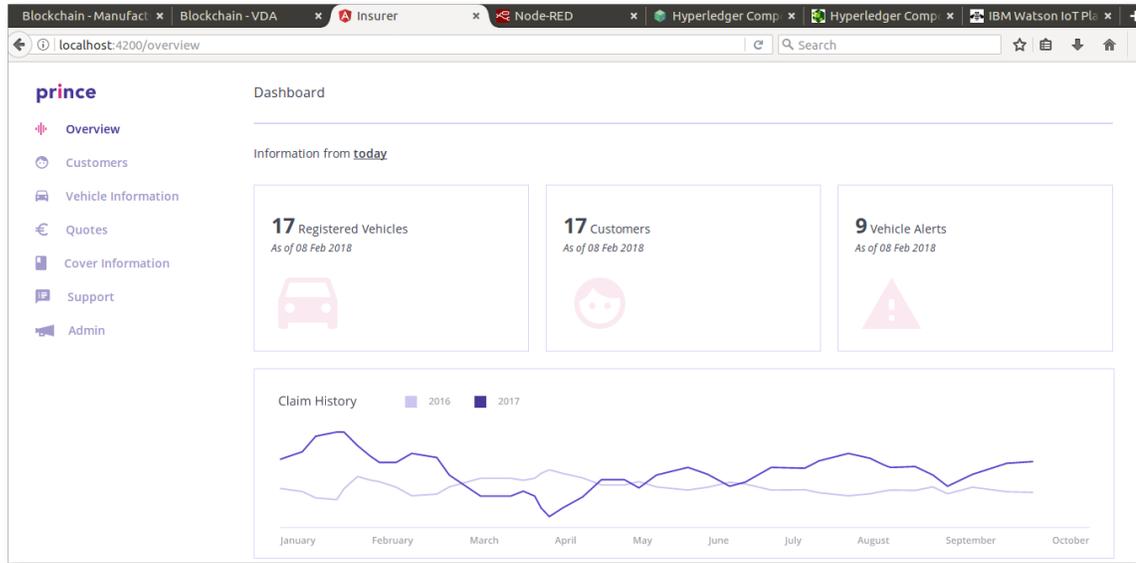


2.3 Insuring the car

As Paul takes ownership of his new car, we will give him the option to insure it. His insurance company offers a discount if he chooses to provide the insurance company with frequent details of the car’s location and other things.

The manufacturer fits the car with a collection of IoT devices, including GPS, air and engine temperature sensors, acceleration information and light information, which can give the insurer information on how the car is being driven, and potentially alert relevant parties if the car is involved in a crash.

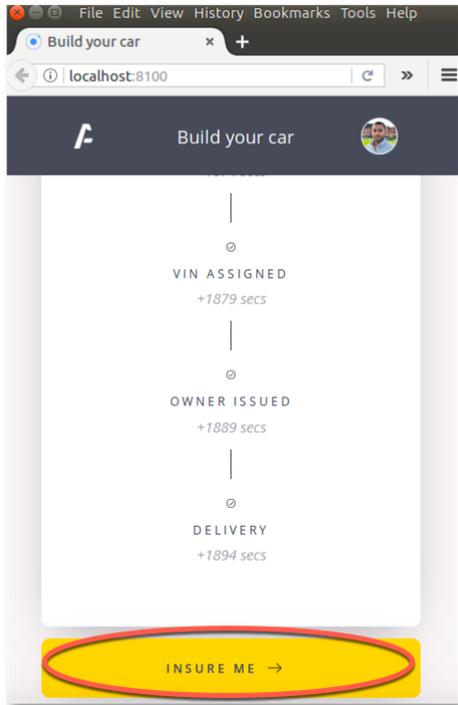
- ___ 13. Switch to the Insurer dashboard (<localhost:4200/overview>). Ensure that the ‘Overview’ tab is selected.



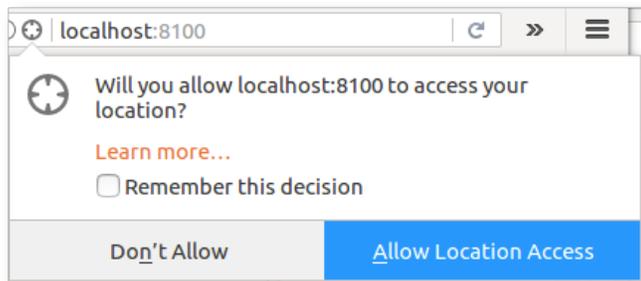
Tommen works for Prince Insurance and this is his dashboard. He requires another subset of information from the blockchain and this view is represented here. He can see information on the cars for which his company is an insurer, and can also approve new policies. (In reality, this latter part can be automated.)

- ___ 14. Switch back to our car buyer’s “Ionic App”. After the car has been delivered scroll down to the bottom, and click the “**Insure Me**” button.

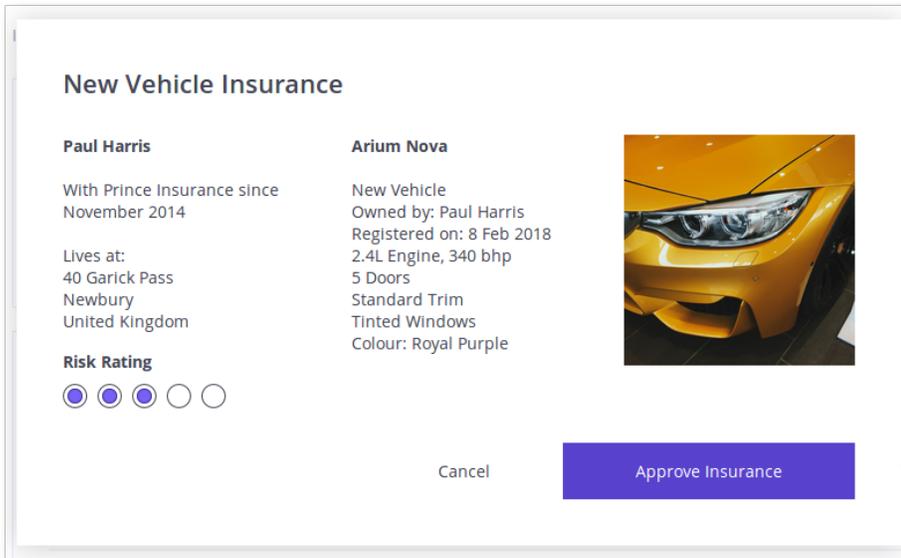
IBM Blockchain



- __ 15. Click **“Allow Location Access”** if a popup appears; Paul is willing to share the IoT device’s location with the insurance company.



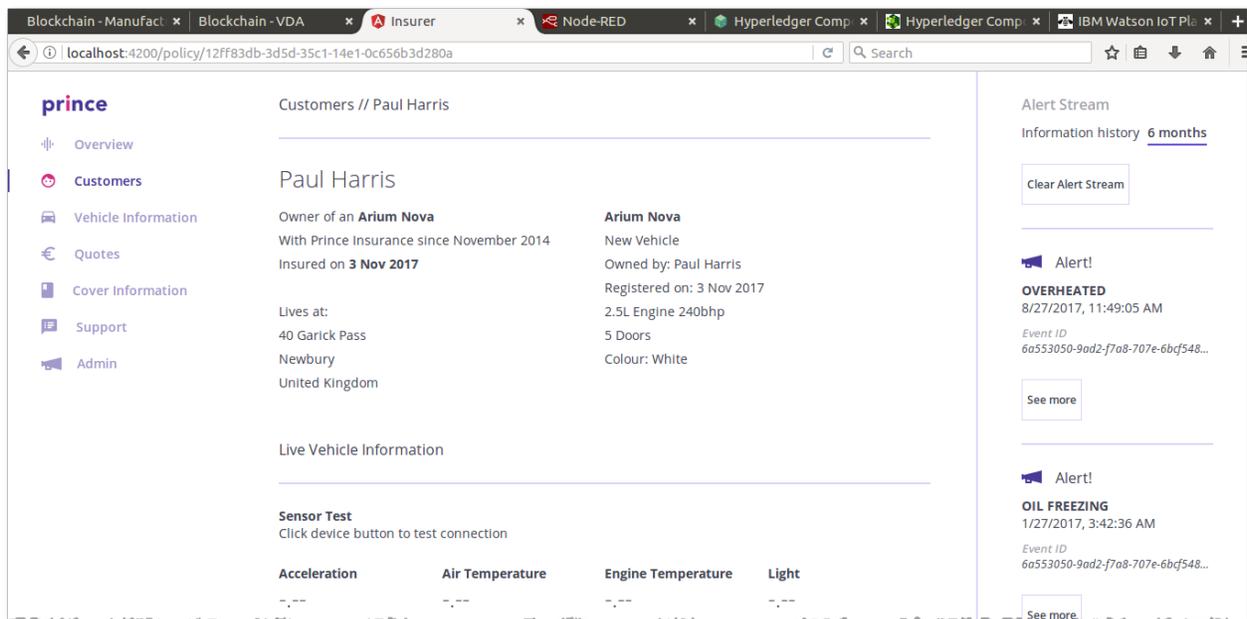
- __ 16. Switch back to the “Insurer” view (<localhost:4200/overview>).
- __ 17. Click the “Approve Insurance” button.



__ 18. Wait for the approval to be logged on the blockchain.

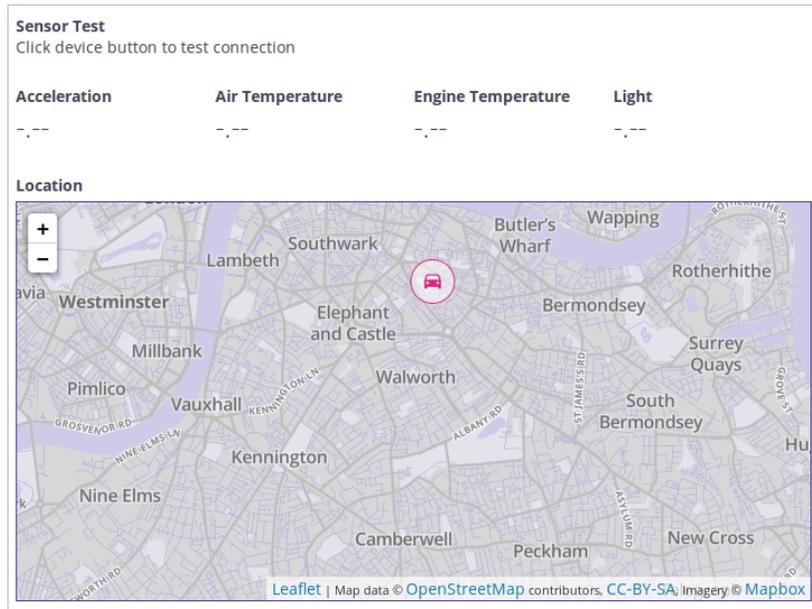
Paul is now insured by the insurance company.

__ 19. Review the 'Customers' tab to see details of Paul's policy.



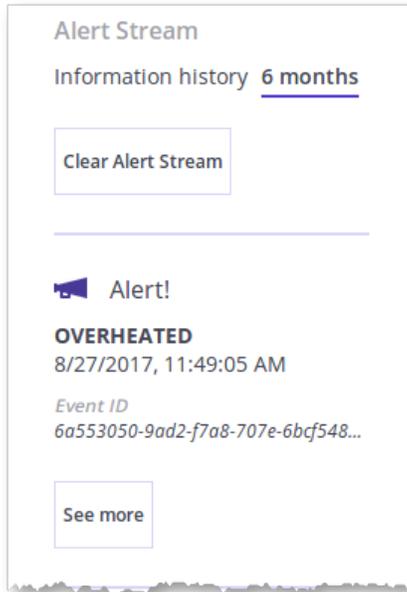
At the top of the page you can see basic details of Paul's policy including his address and car information.

Underneath this is the set of raw readings from the IoT devices attached to Paul’s car. This is useful information for debugging; in reality the blockchain is not used to share complete data streams from the IoT sensors as the amount of data is too great and is not relevant to be shared in its entirety.



However, what would be relevant is the analysis of key events in the IoT stream. For example, if the acceleration is shown to be greater than (for example) 2G, this might indicate a crash event that the insurer might care about.

This is shown as a set of alerts on the right hand side of the insurer’s customer view:



Without a real sensor tag connected into the application, the information displayed here is either blank or mocked up. In the next section we will inject data into the application using internet of things integration.



It is possible to connect a real sensor tag so that its information is displayed in the Insurer view; we have tested using a Texas Instruments SimpleLink Bluetooth SensorTag. To use this, you need to download the TI SimpleLink Starter app to a nearby mobile device, use it to discover the sensor via Bluetooth, note down the unique address of the tag and enable the "Push to cloud" option to submit the sensor readings so that they can be read by the IBM Watson IoT platform. Then you need to update the "IBM IoTP Test Device" node in the Node-RED flow to monitor the readings from the unique address of the tag from the cloud. Remember to redeploy the Node-RED flow.

3 Cool Stuff

IBM Blockchain

In this section, we're going to look at how the scenario can be enhanced by bringing blockchain together with internet of things and analytics.

3.1 Internet of Things Integration

We will start by looking at how sensor data from the car makes its way into the blockchain. To do this we will use an integration tool called Node-RED. This includes a graphical interface to describe how data flows from input sources (e.g. a sensor) to output sources (e.g. the blockchain).

Node-RED has connectors for sending data to, and receiving data from, a blockchain. It also has connectors for receiving data from the IBM Watson IoT platform (for sensor tag integration). We can also generate fake sensor data for testing, in the absence of a physical sensor device.

__ 20. Switch to the Node-RED tab (localhost:1880).

The screenshot shows the Node-RED web interface in a browser window. The browser tabs include 'Blockchain - Manufac', 'Blockchain - VDA', 'Insurer', 'Node-RED', 'Hyperledger Comp', and 'IBM Watson IoT Pl'. The address bar shows 'localhost:1880/#flow/a333211b.340d88'. The Node-RED interface has a sidebar on the left with 'input' and 'output' sections. The main workspace shows a flow named 'IoT Flow' with nodes: 'timestamp', 'Initialise list of allowed events', 'Set Contexts', 'DEBUGGING INPUTS', 'PUSH CONNECT ATTEMPT', 'PUSH OVERHEATED', 'PUSH OIL_FREEZING', 'PUSH CRASH', 'HOT PAYLOAD', 'COLD PAYLOAD', 'CRASH PAYLOAD', 'LIVE IOT INPUTS', and 'IBM IoTP Test Device'. The right sidebar shows the 'info' panel for the selected flow, displaying its name, ID, and status.

| Flow | |
|--------|-------------------|
| Name | IoT Flow |
| ID | "a333211b.340d88" |
| Status | Enabled |

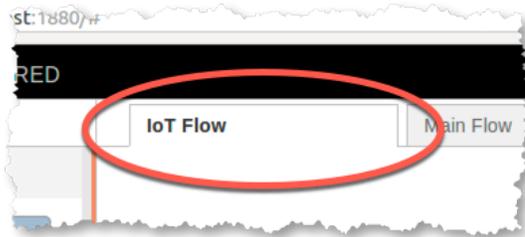
| Information | |
|---|--|
| Your flow configuration nodes are listed in the sidebar panel. It can be accessed from the menu or with | |

The main window shows the flow of how data from devices is mapped to blockchain events. The tabs along the top show the different flows that are deployed. Down the left hand side you can see the available connectors for wiring into the flow. The right hand-side contains the properties of the selected connector and debug information.

IBM Blockchain

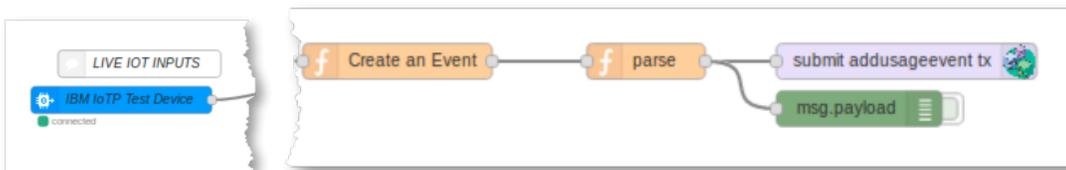
(Note that if you make any changes to the flow, you need to press the “Deploy” button to let them take effect.)

- __ 21. Ensure that the “IoT Flow” tab in Node-RED is selected.



There are some interesting things to note in this flow.

- __ 22. Look at the “IBM IoT Test Device” to “Submit AddUsageEvent TX” flow. This takes readings from a real sensor device and publishes any interesting events to the blockchain.



- __ 23. Double click the “Set Contexts” connector.

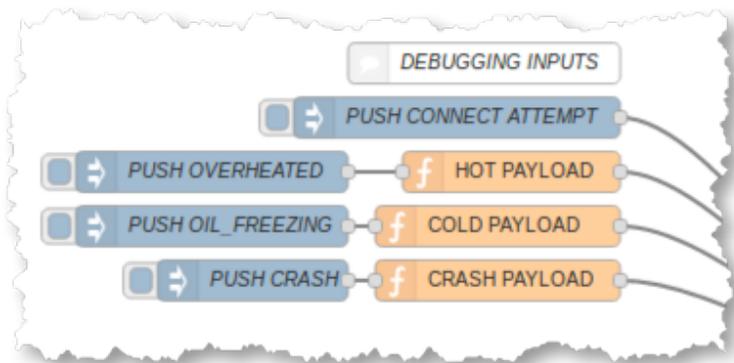
This shows the thresholds for sending interesting events to the blockchain. For example, if the acceleration is greater than 2.2G, this causes a crash event to be sent to the blockchain.

```

node properties
Name Set Contexts
Function
6 flow.set('events', '{"ACTIVATED", "CRASHED", "OVERHEATED"}');
7 flow.set('deviceId', "vehicle");
8 flow.set('pitch', 0);
9 flow.set('roll', 0);
10 flow.set('accSensitivity', 8192.0);
11 flow.set('gyroSensitivity', 65.536);
12 flow.set('dt', 0.01); // 10 ms sample rate!
13 flow.set('too-hot', 100);
14 flow.set('too-cold', 5);
15 flow.set('allowed-acc', 2.2);

```

- ___ 24. Look at the set of connectors next to the “DEBUGGING INPUTS” section: PUSH CONNECT ATTEMPT, PUSH OVERHEATED, PUSH OIL FREEZING and PUSH CRASH.



These connectors allow us to simulate an interesting event occurring, in the absence of a real device.

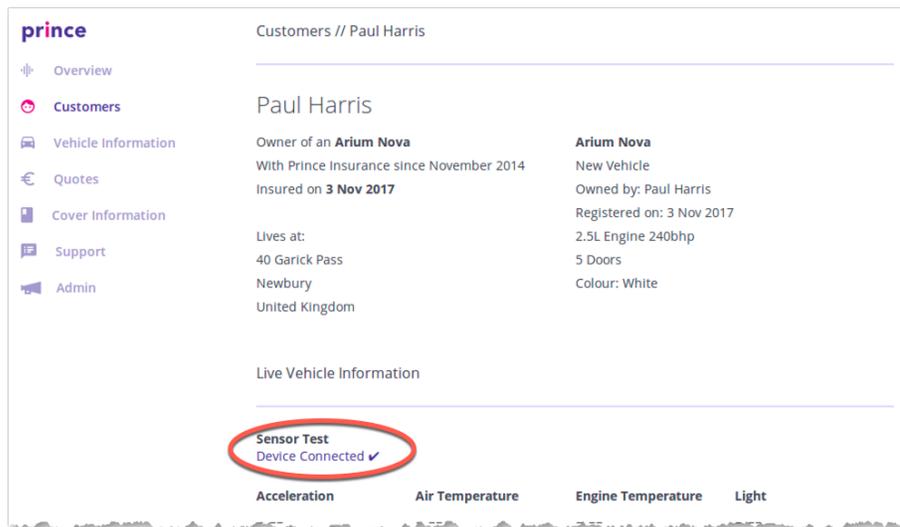
- ___ 25. Click the rounded square button next to the PUSH CONNECT ATTEMPT connector.



You should see a message saying that data was successfully injected into the flow.



- __ 26. Switch to the Insurer tab (localhost:4200), and notice under “Sensor Test” that the vehicle sensor is now connected.

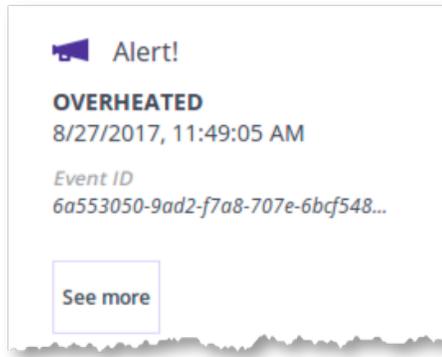


- __ 27. Switch back to the Node-RED tab (localhost:1800), and click the button next to the PUSH OVERHEATED node to send an event to the blockchain which denotes Paul’s engine overheating.



You should again see a “Successfully injected” message.

- __ 28. In the Insurer view you should see an alert that reveals this event to the insurer.



- __ 29. Try invoking the other events too (OIL FREEZING, CRASH) to see their effect.

More details on the IBM Watson IoT Platform can be found on a pre-loaded tab in the web browser

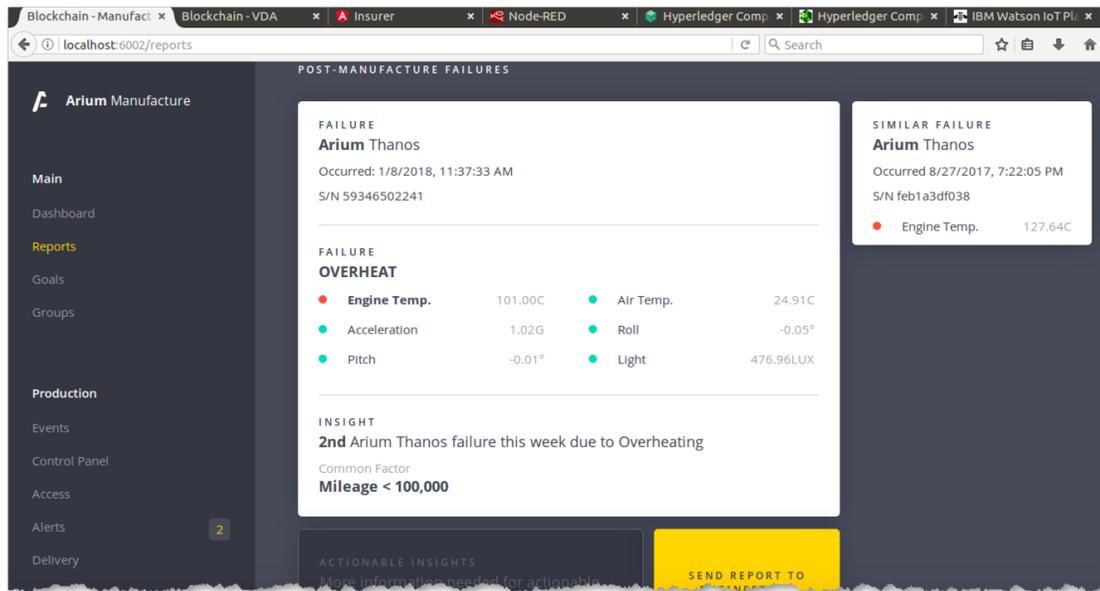
(<https://i5l9uv.internetofthings.ibmcloud.com/dashboard/#/ibmssolanding>).

3.2 Analytics

It is possible to use the information stored on the blockchain to provide insight on aggregate usage patterns to interested authorized parties. This gives the power of data analytics on top of the benefits of a blockchain, as a verifiably clean source of information to analyze.

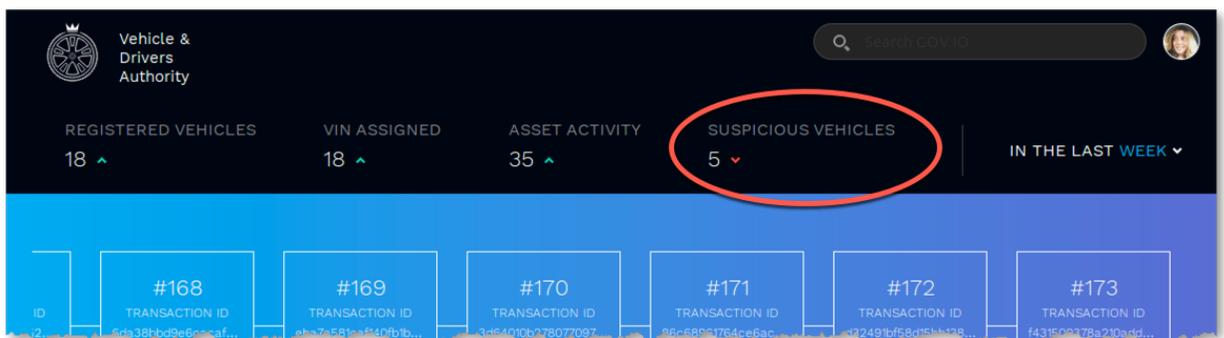
- __ 30. Switch back to the Manufacturer view tab (<localhost:6002>) and click the “Reports” link.

The engine overheated events show in this view. These events were captured in the blockchain and the manufacturer role has permission see this type of event. The manufacturer wishes to detect trends in engine overheated failures in order to determine if a factory defect is causing this condition.



The regulator in this scenario can also run analytics on the transactions on the blockchain to look for suspicious behavior that the smart contract was not designed to prevent from a single invocation.

- __ 31. In the Vehicle & Driver Authority dashboard (<localhost:6001/dashboard>) click the “Suspicious Vehicles” tab near the top.



Here we can see that by performing analytics on the blockchain dataset, we have found a number of vehicles with associated suspicious transactions that may warrant further investigation.

- __ 32. Click on ‘Mileage anomaly’ in the list of suspicious vehicles.

| SUSPICIOUS VEHICLES | | | LAST 24 HOU |
|-----------------------|------------|--------------------------------------|-------------|
| Vehicle | VIN | Notification | |
| Ridge Cannon - White | 312457645 | Suspicious ownership sequence | |
| Ridge Rancher - White | 326548754 | Uninsured vehicle | |
| Morde Pluto - Green | 564215468 | Insurance write-off but still active | |
| Morde Putt - Black | 6437956437 | Mileage anomaly | |
| Ridge Cannon - Silver | 65235647 | Untaxed vehicle | |

This shows a list of the transactions that are associated with this anomaly. In this instance, the mileage of the vehicle may not match with insurance records - or has even has decreased from previous records.

| Morde Putt - Black | | | 6437956437 | Mileage anomaly |
|------------------------|--------------------------------------|-----------|-----------------------|-----------------|
| Timestamp | Transaction | Car Owner | Contact Current Owner | X |
| Nov 3, 2017 6:27:51 PM | b6c71558-0873-335c-635e-adcf32115b17 | Anastasia | | |

4 Next Steps

In this lab you have experienced a live blockchain solution through the eyes of four participants of a vehicle network: a buyer/owner, manufacturer, regulator and insurer. A blockchain can be used to great effect in this business network because there is a clear need to share information and value in participants being able to trust the information they see.

Where you go from here is up to you.

1. **Important!** Be sure to cleanup the hyperledger fabric environment for subsequent labs. Perform the following at the command prompt in the VLD directory:

IBM Blockchain



- a. Open a terminal window right clicking on the terminal icon  on the left-hand navigation pane of the Ubuntu Linux desktop. Select **New Terminal** and a new terminal window will appear.
 - b. **cd VLD**
 - c. **./stopAll.sh**
 - d. **cd ~**
- __2.

If you have a technical background, consider finding out more about Hyperledger Fabric and the IBM Blockchain Platform development experience. For no charge you can sign up to the IBM Blockchain Platform 2.0 Beta Plan to play more with the blockchain technology and implement your first use-case.

If you are interested in the potential benefits of blockchain in your business, IBM has a bunch of services that can help. Start by going to www.ibm.com/blockchain.

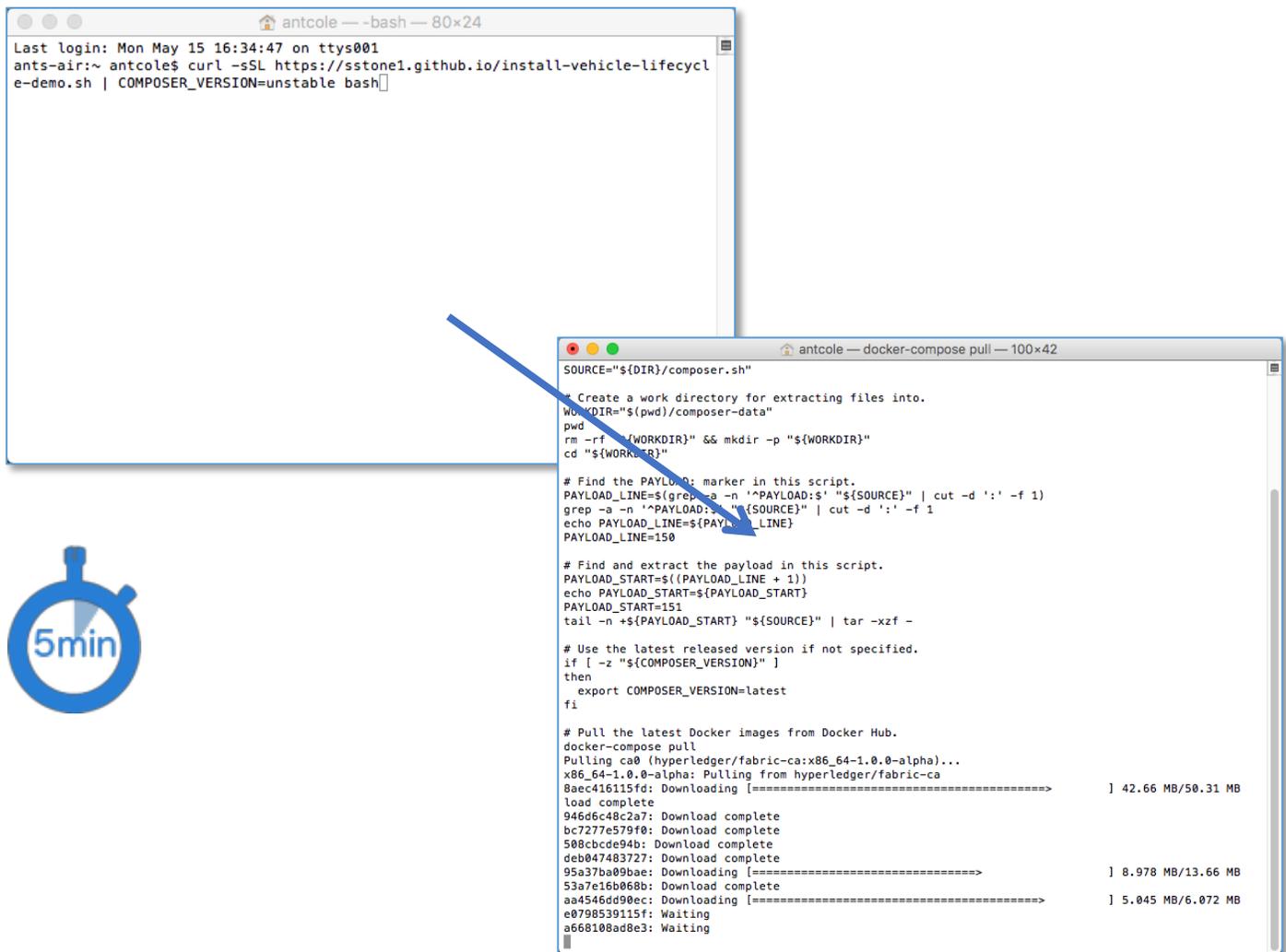
Congratulations on completing the lab!

Appendix A. Restarting the sample application

The virtual machine used in the lab is set to start the application, blockchain and associated services automatically. If for any reason, they have broken or need restarting, open a terminal and type:

```
cd Documents/composer-sample-applications-master/packages/vehicle-lifecycle
./build.sh
cat installers/hlfv1/install.sh | bash
```

The application should take around five minutes to restart.



```
antcole — -bash — 80x24
Last login: Mon May 15 16:34:47 on ttys001
ants-air:~ antcoles$ curl -sSL https://sstone1.github.io/install-vehicle-lifecycle-demo.sh | COMPOSER_VERSION=unstable bash

antcole — docker-compose pull — 100x42
SOURCE="${DIR}/composer.sh"

# Create a work directory for extracting files into.
WORKDIR="$(pwd)/composer-data"
pwd
rm -rf "${WORKDIR}" && mkdir -p "${WORKDIR}"
cd "${WORKDIR}"

# Find the PAYLOAD marker in this script.
PAYLOAD_LINE=$(grep -a -n '^PAYLOAD:$' "${SOURCE}" | cut -d ':' -f 1)
grep -a -n '^PAYLOAD:$' "${SOURCE}" | cut -d ':' -f 1
echo PAYLOAD_LINE=${PAYLOAD_LINE}
PAYLOAD_LINE=150

# Find and extract the payload in this script.
PAYLOAD_START=$((PAYLOAD_LINE + 1))
echo PAYLOAD_START=${PAYLOAD_START}
PAYLOAD_START=151
tail -n +${PAYLOAD_START} "${SOURCE}" | tar -xzf -

# Use the latest released version if not specified.
if [ -z "${COMPOSER_VERSION}" ]
then
  export COMPOSER_VERSION=latest
fi

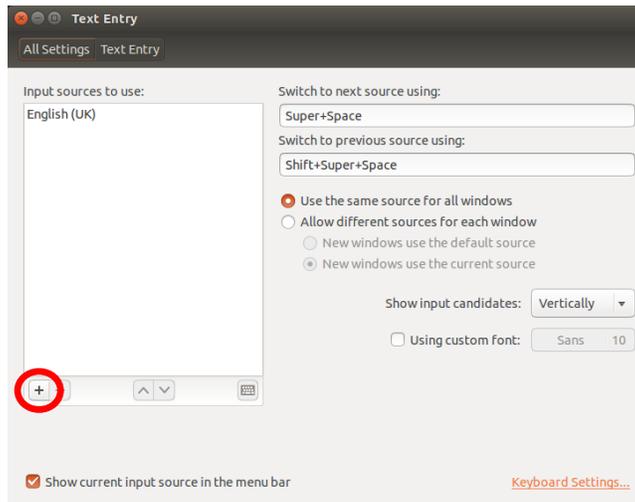
# Pull the latest Docker images from Docker Hub.
docker-compose pull
Pulling ca0 (hyperledger/fabric-ca:x86_64-1.0.0-alpha)...
x86_64-1.0.0-alpha: Pulling from hyperledger/fabric-ca
8aec416115fd: Downloading [=====] 42.66 MB/50.31 MB
load complete
946d6c48c2a7: Download complete
bc7277e579f0: Download complete
508cbcd94b: Download complete
deb047483727: Download complete
95a37ba0b9ae: Downloading [=====] 8.978 MB/13.66 MB
53a7e16b068b: Download complete
aa4546dd90ec: Downloading [=====] 5.045 MB/6.072 MB
e0798539115f: Waiting
a668108ad8e3: Waiting
```



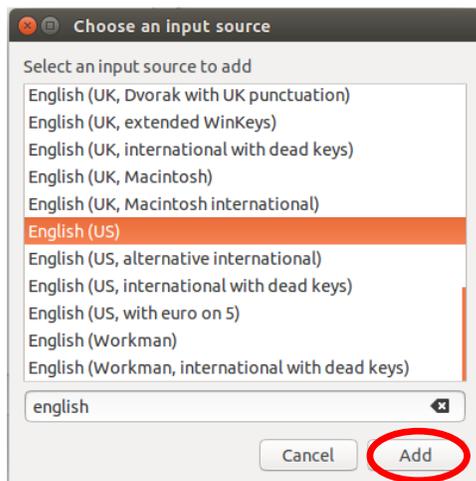
Appendix B. Changing the keyboard language

To change the keyboard language to enable you to use laptops from another country follow these steps:

- ___ 33. Click on the  icon in the top right & select **Text Entry Settings...**
- ___ 34. **Select** the  symbol.



- ___ 35. **Type** your **Language** and then **country**.
- ___ 36. **Select** the appropriate keyboard and click **'Add'**



IBM Blockchain

- __ 37. **Close** the Settings box.
- __ 38. Select the '**EN**' in the top right of the screen and select your new keyboard.



- __ 39. Your keyboard is now ready to use.