Automated & Timely Crop Insurance Payments

Farmers in developing countries like India have always been at the mercy of the weather. Changing weather patterns have further aggravated the problem. When a natural calamity strikes, farmers need help immediately! One way to ensure they get help quickly when disaster hits, is to provide insurance to farmers at an affordable cost.

Easy, trusted and affordable crop insurance is therefore crucial, not just for the well-being of farmers but also for the food security of the nation. Traditional forms of crop insurance are slow, capital-intensive and often economically nonviable for private businesses. Farmers are also unwilling to get their crops insured because of a lack of trust in the insurance companies and the fear of delayed/non-payment of claims.

An open, trusted, low-cost and automated crop insurance system will ensure that many more farmers come under the insurance umbrella and benefit from timely crop insurance payments.

The cost of administering insurance is currently a significant barrier to entry into this space. This cost can be reduced dramatically with the right use of technology.

It will open up a vast new market for efficient and competitive private businesses while at the same time addressing a vital social need and limiting the need for government intervention.

**Technology:**

The Icertis AI/ML and Blockchain Hackathon is an Azure Cloud-only Hackathon, with all solutions to be built and evaluated on the Azure Cloud.

* Azure Free Trial and Developer Resources: [**Azure Developer Resources and Azure Free Trial Signup**](http://www.aka.ms/hack.azure)
* Please plan your free trial periods so that your submission is active for evaluation till the 15th of December.
* You might have to create multiple free trial accounts in a staggered fashion.

**Note:**

* The following are meant to be guidelines to provide direction to teams. However, all teams are encouraged to take their solution above and beyond and come up with solutions that best address the provided use case.
* In case of uncertainty, please document your assumptions and proceed based on your best judgement.
* If you unable to complete some part or find yourself stuck, use your discretion to come up with mockups and work arounds.
* Your problem-solving ability is one of the factors on which your submission will be evaluated.

**Objectives:**

* Provide an easy to use, secure interface to manage insurance policies, farm details and other information through a mobile app/web site.
* Use block chain technology to design a solution that will allow a secure, technology based, and fast settlement of crop insurance claims.
* Integrate weather data to determine drought or excessive rainfall at farm locations to automatically trigger settlements.

**Methodology:**

1. User interface: Provide an easy to use interface to gather farmer personal data, farm coordinates and crop information for a particular year. Build responsive UI web pages that will also be rendered correctly on an iPhone or an Android phone browser.
   * Stretch goal: build a mobile app on the iPhone or Android.
   * Stretch goal: support one more language in addition to English as the UI and the insurance policy document.
2. The web page/app should allow a farmer to take a picture of his/her insurance policy and upload it.
3. When the insurance policy is uploaded, the following information should be automatically extracted from the uploaded image (insurance policy examples that you can use to test will be provided) Unique policy number, Amount Insured, Start Date, Expiry Date and Geo coordinates of the farm as vertices of a polygon.
4. The extracted information, after it is validated by the farmer, should be written to the blockchain with the insurance policy document as a smart contract as described below.
5. Create a blockchain using [**Blockchain as a Service (required)**](https://docs.microsoft.com/en-us/azure/blockchain/service/Azure)
6. Set up a smart contract : The smart contract should consult an Oracle, and if the insurance policy is in effect (date between start date and expiry date), check if the geo-coordinates are within the areas marked as under drought by the Oracle service (the Oracle service will return an entire set of polygonal vertices of areas under drought)
7. Integrate and analyze real-time weather data with analysis of weather conditions for farm coordinates.
8. Determine whether weather conditions at the farm coordinates in a given period have resulted in a drought or excessive rainfall. If they have, determine the amount to be paid to the farmer based on the insurance policy, weather severity and duration, crop type, crop status and farm location. A simple illustrative calculation could be:
   * In the 6 month period, (drought condition at farm location > 15 days), crop = rice, farming period = seed sowing) results in payment of 100% of amount insured.
   * If the drought condition at farm location < 7 days, the payment could be 25% of the amount insured.

 The farm will be said to be under drought if it is fully enclosed within the area under drought from the Oracle. If the farm is under drought, the smart contract should execute to release a payment automatically to the farmer. A smart contract solution using Azure Blockchain between farmers, weather data providers, insurance company and technology solution provider.

 Allow for participation of insurance providers and regulators in the system through the block chain.

 Design a secure and credible system with the right use of technology.

 These are meant to be broad guidelines for teams to get going. Teams are encouraged to take their solution above and beyond and come up with solutions that best address the provided use case. In case of uncertainty, please document your assumptions and proceed based on your best judgement.

**Team Evaluation Procedure/Criteria:**

Submissions will be evaluated on the following, not in any particular order:

1. Working UI, Blockchain and smart contract implementation.
2. It works end to end!
3. Accuracy of data extraction
4. Accuracy of drought detection
5. User experience
6. Innovation
7. Scale and performance
8. Extensibility
9. Efficient use of Azure resources
10. Team work

**Deliverables To be Given By the Candidate:**

1. Azure solution url. (Mandatory)
2. How Azure have been used in your submission. (Mandatory)
3. An overview and architecture PDF / PPT. Please find the [**(SAMPLE PRESENTATION)**](https://techgig.com/files/contest_upload_files/Icertis_Final_PPT.pptx) which you need to follow for submission. (Mandatory)