Group 10: Automatic geo-location with KasaDaka Weather Service

Riccardo Bianchi	David Huser	Tim Koning	Shruti Rao
2633151	2674972	2595660	2636454

0. Name

Rainfall and storm forecast with automatic geo-location

C: Changelog

#	What was updated?	What are updates based on?
2.1	Relevant Literature	Assignment requirement to state some relevant previous work
4.1	Value Exchange Model	Assignment requirement for expansion
6.1	Interaction view	Updated to show increased fidelity via network resilience, performance improvement via caching, and error voice response Added functionality based on A2 feedback: time-window selection
10.1	Sustainability	Separated feasibility from sustainability. Reframed sustainability based on four types [11]
11.1	Key Requirements	Assignment requirement
12.1	Prototype Description	Updated the design decisions to reflect decisions behind time-window selection. Updated Frontend/backend results and added Data Model
15.1	Usage Scenario	Updated usage scenario to account for additional application functionality
16.1	Feedback Questions	Updated to reflect response to feedback on Assignment 2
17.1	Scope and Fidelity	Assignment requirement
18.1	Conclusion	Added to summarise the application and state future extension

1. Summary of key idea

Understanding rainfall and incoming storms are crucial for farmers in Sahel who are often unable to make timely decisions regarding their crops due to unpredictable weather, coupled with a shortage of relevant, accurate weather alerts [1]. KasaDaka is a voice based information system that can be accessed by farmers by calling to request precipitation data for their region [2]. Currently, users are required to specify the location they would like to obtain rainfall data for, which is often their current location [1]. This proposal aims at improving user functionality by suggesting GPS based auto detection of the users' location.

Due to the increasing inaccuracy of the forecast for one location depending on the terrain and distance to the physical hardware, multiple systems need to be deployed in multiple locations to increase geographical coverage of the service. This, in turn, has implications beyond technology like business and sustainability considerations. The service proposes clustering villages based on their geographical location. A cluster should aim to include areas with good and zero network coverage. For each cluster, a KasaDaka unit with GPS will be installed in an area with good network coverage to which users belonging to the cluster can call. Assuming multiple KasaDaka platforms allows the service to cater to better identifying user location and overcome the problem of requiring a reliable Internet network.

Neighbouring countries of Ghana, Burkina Faso, Mali and Cote d'Ivoire were selected for the initial implementation of the service [4]. Ghana and Burkina Faso have a larger network coverage while Cote d'Ivoire and Mali have sparse coverage (Appendix: Figure 9). Data collected from areas with coverage can thus be shared with users in low or no coverage areas (Appendix: Figure 8).

The service also notifies callers in case of an upcoming storm. Using Human Computer Interaction (HCI) ideals, the proposal wishes to apply a similar ideology and focus on a service that provides ease-of-use for rural users by masking complexity and giving them access to rainfall and storm information with minimal interaction [6]. This added simplification can improve interaction between the users and the service, allowing for it to be accepted with greater ease. Shorter calls imply less money spent by end-users. Finally, one-step access to information can prevent strain on the basic infrastructure and minimize the need for having to train users to use the voice service.

2. Relevant Literature

A few projects have used voice technology to develop applications however, most have remained exploratory or part of pilot studies. A Session Initiation Protocol (SIP) based Interactive Voice Platform service using weather was hosted on the Mobi4D platform [13]. This service made use of messaging services to allow users to send text messages with keywords and hear weather data as voice messages [13]. However, this tool relied on users being able to read and type messages. Based on the same Mobi4D platform, an extension was developed that used Automated Speech Detection to recognise and learn user languages and provide weather updates via a human-like conversation [14]. While an ambitious attempt, the project might be unsuitable for real world deployment due to the requirement of sophisticated processors for running algorithms. Mr Meteo was another tool developed using voice technology to provide weather data to rural users in Ghana, Africa [12]. It relies on the KasaDaka platform and uses user input of their location to provide weather data [12].

3. Actors and goals

Stakeholder	Operational goal	Responsibility in the envisaged system
Farmer	Gain information regarding rainfall for the upcoming week and be informed of upcoming storms	● Use the service to request rainfall prediction ● Heed storm notifications provided by system ● Share awareness of the service with other farmers
System developer	Develop a usable service that meets all Key Requirements (Must Have, Should Have, Could Have) based on stakeholder needs	● Develop a service that requests weather data from external weather API ● Provide farmers with weather data and storm alerts in spoken language at their location ● Ensure that service meets the requirements of the stakeholders
Weather data service	Gather accurate weather data and maintain API	Provide timely and accurate weather data to the database
Internet provider	Ensure internet access and coverage across the region	● Maintain a reliable 3G internet service ● Ensure sufficient coverage and network quality for service to obtain data without any lags [3]
Telephone provider	Ensure GSM network coverage across the region	● Allow for consistent network to allow for the service to be accessible by phone ● Ensure maximum network coverage to allow

		for the service to be used by all ● Provision affordable telephone plans to allow for rural penetration
Non-Government Organi-zations (NGOs)	Provide monetary assistance for purchase of hardware required to set up the service across multiple locations. Also, act as a bridge between local communities and the service	● Provide funding to allow for the KasaDaka infrastructure to be set up in multiple locations ● Attract and encourage local sponsors to invest in the service ● Work with local communities to encourage and support the use of the service ● Act as an intermediary between local communities and the service to convey shortcomings, and problems faced by them

4. Context and scope

Stakeholder concerns: a *farmer's* concern is to know the weather (rainfall, storm alert) for the next few days in order to plan work or protect crops, thus the farmer is interested in receiving accurate forecasts for each day in an understandable and accessible way. This means the information should be delivered in the preferred spoken language via the technology closest to people in the target countries: a basic or feature phone.

The *system* itself will detect its geolocation. The *weather data service* is accessed by the system for the weather forecast. The system should translate it into the requested language as localized text-to-speech (or data-to-speech) for the farmer.

A *system developer* installs and maintains the system, e.g. set up the access of the system to the weather data service, and provides demos or training for farmers.

NGOs supply hardware to the developer and are the main cost carrier of the devices and maintenance. The related context view is visualized in Figure 1.

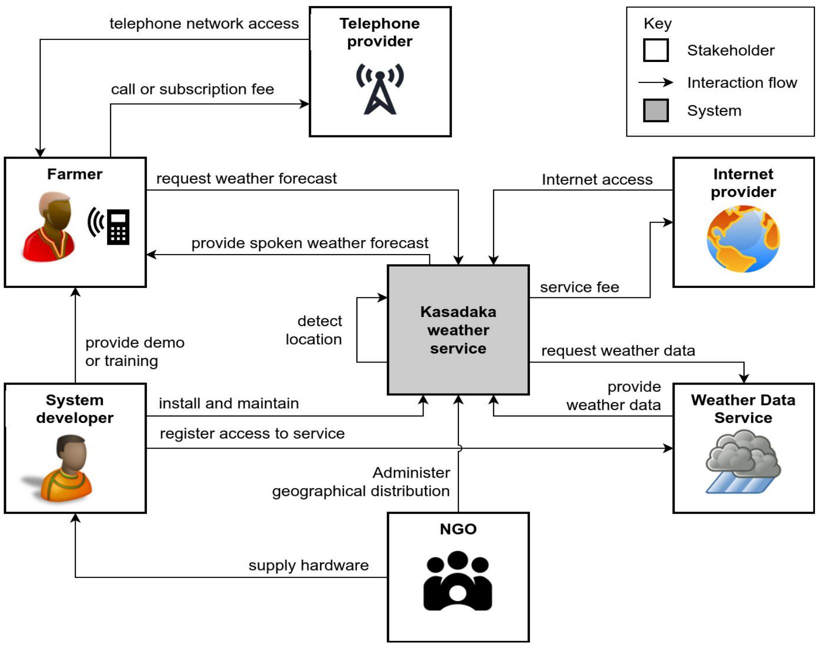

Figure 1. Context view

For a **successful usage of the system**, the farmer must be able to access the system by calling a telephone number at any given time. The received forecast must be sufficiently accurate for the farmer's current geolocation and available languages and the system's transmitted audio must be understood by the farmer in order to act on this information. Moreover, the system must be usable by multiple farmers. A system developer and the NGO are interested in easy installation and maintenance, as well as a preferably free weather data service to keep running costs low.

Assumptions are that farmers possess a mobile phone with the ability to access the telephone network. They know how to access the system, or alternatively are responsive to training. Farmers are also aware of the nearest KasaDaka due to geographical deployment. The system is composed of multiple KasaDakas, where each needs to connect with both GSM (3G network to access the Internet) as well as GPS and must be connected to a reliable electricity source (solar or power grid) and resilient to intermittent power outages. The farmer must be (physically) close to the deployed system in order to provide accurate forecasts.

5. Value Exchange Network for Stakeholders

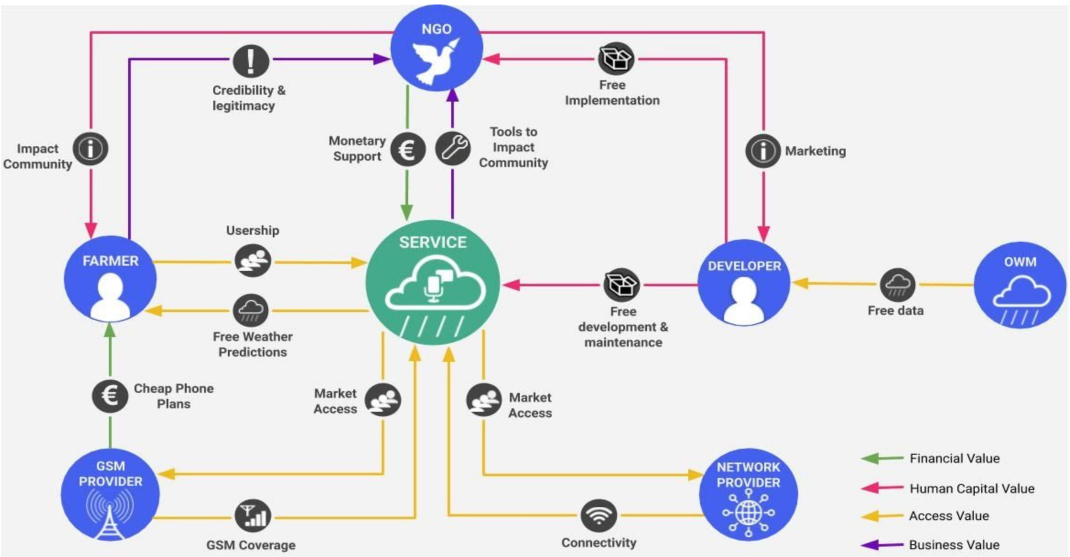

Figure 2: Value Exchange between Stakeholders

Figure 2 shows the various stakeholders and their means of contributing and benefitting from each other within the service ecosystem. Top level business values were identified and associated with each stakeholder interaction pair. Often, an interaction pair had more than one type of value, however only the most significant one was depicted. For example, OWM provides *access* to free weather data to the developer which in turn results in a financial gain. But only the benefit of *access* is highlighted.

6. Use case scenario script

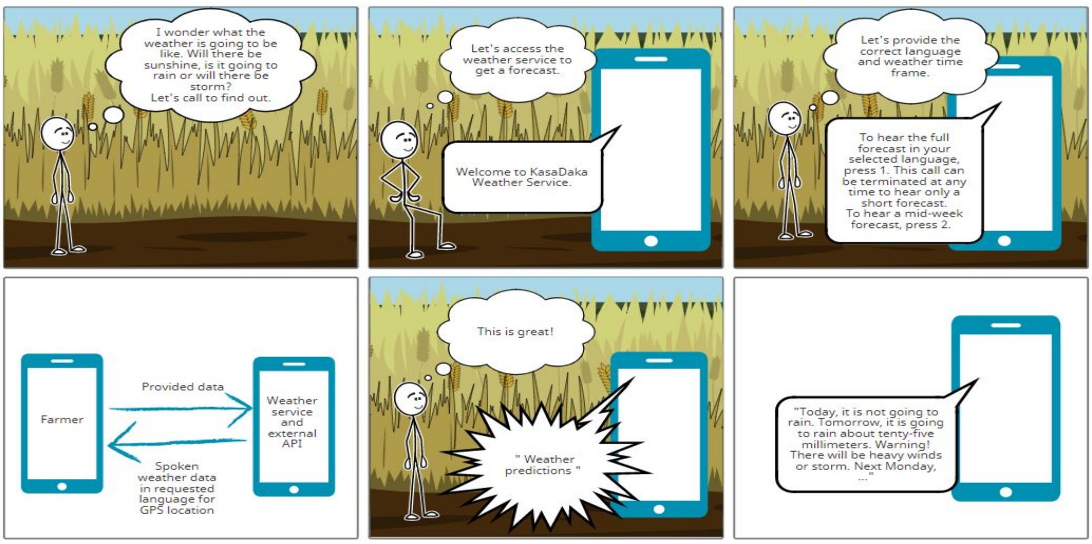

Figure 3. Storyboard view

7. Interaction and communication

Figure 4 provides a UML-based description on how farmers interact with the system and the logic for the obtainment of the forecast and geo-location and related caching.

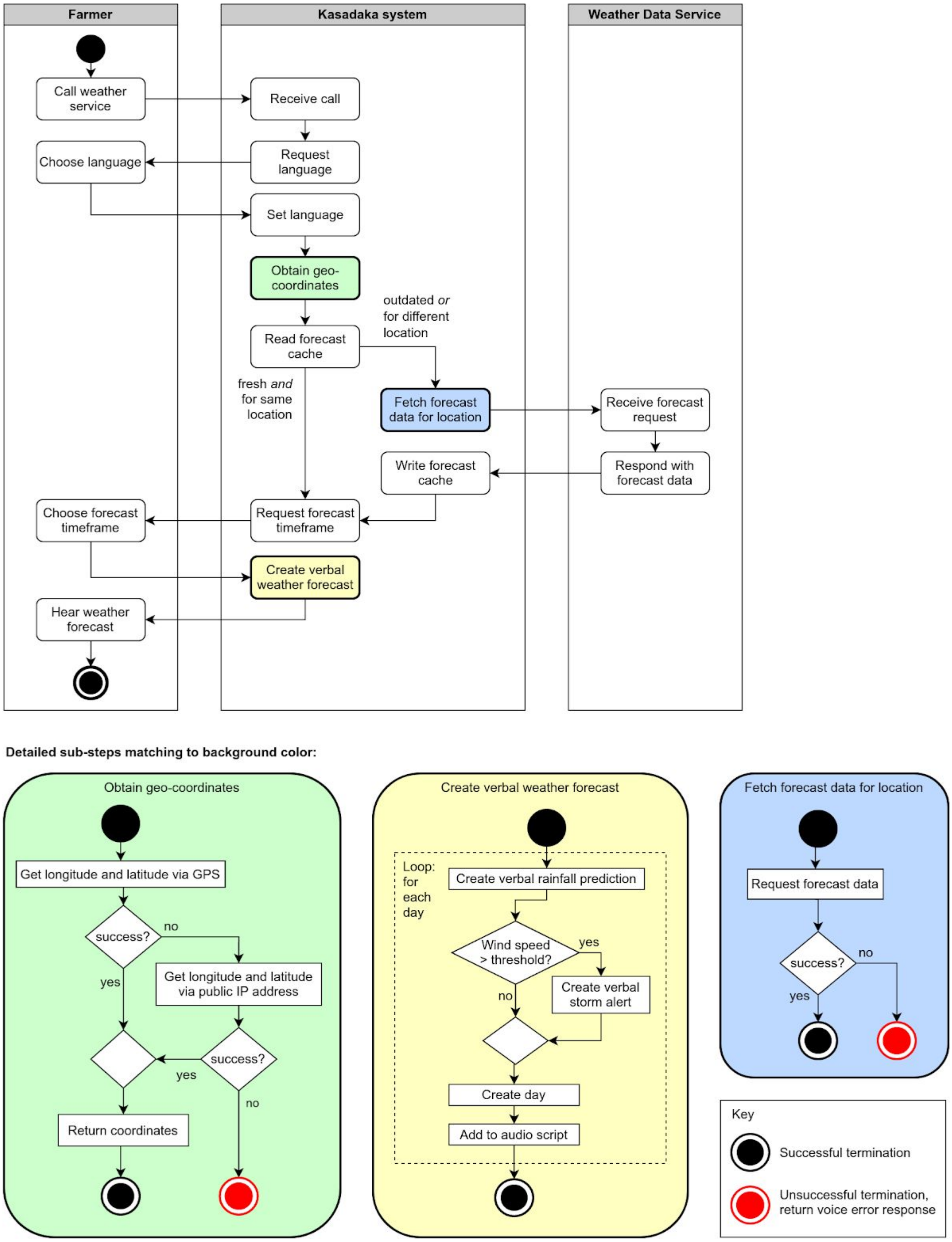

Figure 4. Interaction view

8. Information concepts

Figure 5 provides a UML-based description on how to construct spoken weather forecasts based on geographical and weather information resources and selected language (cf. Figure 4). The actual implementation class diagram is found in Figure 7.

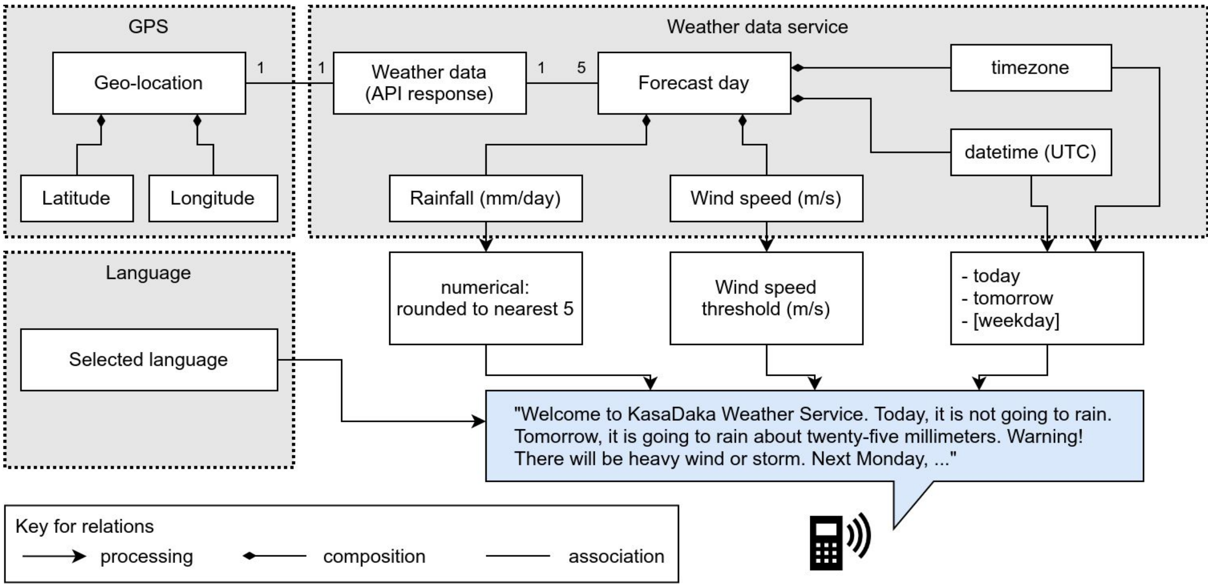

Figure 5. Information concept view of voice weather forecast

9. Technology infrastructure

Figure 6 shows the relationship between the hardware and software components that are involved in order for the service to work in tandem with KasaDaka infrastructure.

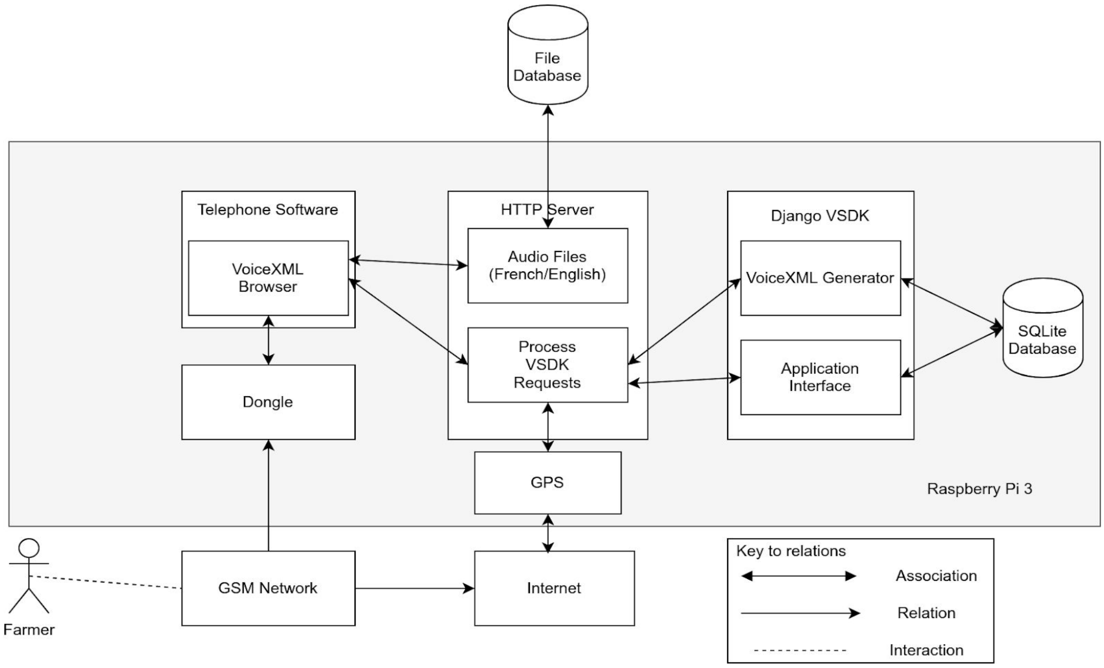

Figure 6. Technology Infrastructure

10. Cost considerations

Category	Description	Cost reduction methods	Cost carrier
KasaDaka Hardware (Raspberry Pi 3, GSM 3G Dongle, Real Time Clock module, GPS & Solar module)	Hardware costs for hosting the service.	Use hardware that is cheap and available in-country.	NGOs to enable multiple installations of KasaDaka

Development	Cost of the development of service consisting of salaries and software licences.	Use free and open-source software.	Volunteers from VU ICT4D course
Hardware maintenance	Cost of repairing and replacing hardware.	Select hardware that is reliable and can be replaced easily and at low cost.	NGOs
Software maintenance	Cost of developers to fix problems and change software to the changing needs of farmers.	Maintaining modularity regarding the software components to easily switch out components.	NGOs
Infrastructure	Cost of using telephone networks, internet connectivity and GPS data location. Not including the calling costs from farmers.	Service platform that is independent from the telephone infrastructure. Engage local businesses and carriers to provide cheap plans.	NGOs

11. Feasibility and sustainability

In terms of feasibility, the minimum hardware requirements matching the described specifications or versions will be required for the product to function as expected. For the service to remain feasible, GPS has to be installed, configured and tested to work with the service across multiple clusters. Frequent maintenance of hardware and software will be required to ensure continuous service. Challenges may arise from finding volunteers with the required level of expertise to manage the technical aspects. Since a single KasaDaka with GPS is not expensive, attracting capital for its deployment should not be a challenge. Overall, based on the feasibility assessment and minimum requirements for users, the service is deemed viable for use within the identified regions and also expandable to service users in other regions of Africa, or other countries with similar socio-economic problems.

Four categories pertaining to technical, economical, social and environmental sustainability were identified [10]. Maintainability of hardware and software in the long run remains the main concern for *technical sustainability*. Additional steps will have to be taken to protect hardware from environmental, theft and property damage. To preserve capital and ensure minimal costs, external help would be required to enable *economic sustainability.* Raspberry Pi Foundation donates hardware across South Africa and could be convinced to provide a free initial supply of devices [5]. The caching mechanism prevents a linear increase of costs of using OWM when usage increases. *Social sustainability* should allow for support of local communities and activities that generate benefits for such communities. The service would have to rely on NGOs and businesses for donations to keep costs to a minimum. Local communities might not trust the service due to long rooted traditions and will have to be convinced that they can rely on the service and use it for decision making through the help of the system developers and local volunteers. Without sufficient users and assurance of community acceptance, it will be challenging to get businesses to invest in the service. Radio stations are an established channel of communication that can be used to sustainably promote the weather service without significant marketing cost. Further *social sustainability* ideas could include local businesses. However, they will require having interest in local economic progress to share the incentive to get involved. NGOs might have to act as a bridge between such businesses and the service. The service also aims for *environmental sustainability* through ensuring energy efficiency and minimum carbon footprint through use of Solar modules. The overall setup also involves a single board computer that has low energy consumption and low maintenance.

12. Key requirements

For the project, *Must Haves* formed the Minimal Usable Subset of requirements that were needed at the very least to use the service [9]. *Should Have* requirements were deemed not vital to the service however, would greatly enhance the service quality [9]. *Could Have* were all desirable requirements that would improve user and developer experience but leaving them out would not have any noticeable impact on the service [9]. *Won't Have* requirements were nice to have but were not be added due to project scope [9]. All requirements under *Must Have, Should Have and Could Have* formed the final project. Based on the scope of the assignment, categories were also identified and defined for the Software Requirements Specifications (SRS) under Design Constraints **(D)**, Interfaces **(I)**, Functionality **(F)**, Required Performance **(P)** and Quality Attributes **(Q)**.

#	Requirement	Priority
D 1.0	Service must be set up with Raspberry Pi 3, GSM 3G dongle and Real Time Clock module to connect to the respective networks	Must Have
D 1.1	Service should have a GPS module to enable auto-geolocation detection	Must Have
D 1.2	Service should be connected to a solar power module for power supply	Must Have
D 2.0	User should be able to hear and understand the voice messages in their preferred language	Must Have
D 3.0	Service must be registered with Open Weather Map API	Must Have
I 1.0	Service must ask caller to choose between English and French as their preferred language of interaction	Must Have
I 1.1	Service must not ask repeat callers to choose a preferred language of interaction	Could Have
I 2.0	Service should ask the user to choose between hearing full forecast or mid-week forecast	Should Have
I 3.0	At the end of the message, Service should ask the user if they wish to hear the message again	Won't Have
I 4.0	Service should have a web interface for NGOs to access and view maps of villages covered by the service	Won't Have
I 5.0	Service should allow user to choose a different location and hear predictions	Won't Have
F 1.0	Service should detect its location using installed GPS	Must Have
F 2.0	Service must remember the language choice for repeat callers	Could Have
F 3.0	Service should make daily requests to the Weather Data Service (Open Weather Map)	Must Have

F 4.0	Service should be able to obtain rainfall and storm predictions for upto a week in advance	Must Have
F 5.0	Service should be able to generate spoken rain and storm predictions for today, tomorrow and at least the next five days	Must Have
F 6.0	Service should check for strong wind speeds daily	Should Have
F 6.1	Service should generate storm alerts only in case of winds exceeding a certain threshold	Should Have
F 6.2	Service should have option to change storm thresholds manually	Should Have
F 7.0	Service should inform the user of the location that the predictions are relevant for	Should Have
F 8.0	Service should present rainfall and storm predictions for the number of days selected by the user	Must Have
F 9.0	Service informs users of the type of storm based on additional weather parameters	Won’t Have
F 10.0	Service should provide additional crop information associated with the rainfall prediction	Won’t Have
F 11.0	Service should gather feedback from users regarding accuracy of predictions	Won’t Have
P 1.0	Service should have a local cache for weather data in case of no internet access	Should Have
P 1.1	The local cache should be refreshed without user intervention	Should Have
P 2.0	Service should obtain public IP geo-location by making HTTP calls to a free web service in case of no GPS	Must Have
Q 1.0	Service should be accompanied by a README and technical documentation	Should Have
Q 1.1	Service should provide documentation to extend and add new languages	Could Have
Q 2.0	Service should provide error message to user in case service cannot retrieve weather data or access local cache	Should Have

13. Prototype description

Design Decisions: The application implements a language selection and asks the user to choose between listening to a five day forecast or a mid-week forecast. A five day forecast was defined to include forecasts for today and the following four days. A mid-week forecast includes forecasts starting the day after tomorrow. It was identified that land farming tasks can be classified as land preparation and land management, with most daily work evolving around land management [10]. Land management involves shorter, frequent days of activity to ensure soil conditions are maintained and typical activities include fertiliser preparation, field workability and irrigation decisions, all of which require knowing rain conditions a only few days in advance [10]. Thus, farmers might find less value in next day forecasts which leaves them with insufficient time for useful preparations. Mid-week forecasts presented by the service however could be used to assist with land maintenance. By presenting this additional choice, the application allows user preference and requirements to dictate the length and content of the voice message. The forecast message was made concise by presenting wind information only in case of a storm. Wind speed threshold was made adjustable within the administration backend because wind speeds might be perceived differently in different regions. Given that the information medium is via voice, it is imperative to recognize human limitations for processing long messages within a short period of time [7]. By keeping the message as short as possible, users are less likely to be overwhelmed by the content and forget what was said [7].

Data Model: Extending the Information concept view (Figure 5), the class diagram in Figure 7 displays the actual implementation of the voice service, utilizing the KasaDaka-VSDK as the backbone for rendering the weather information via Voice Labels (audio files) into a Django template.

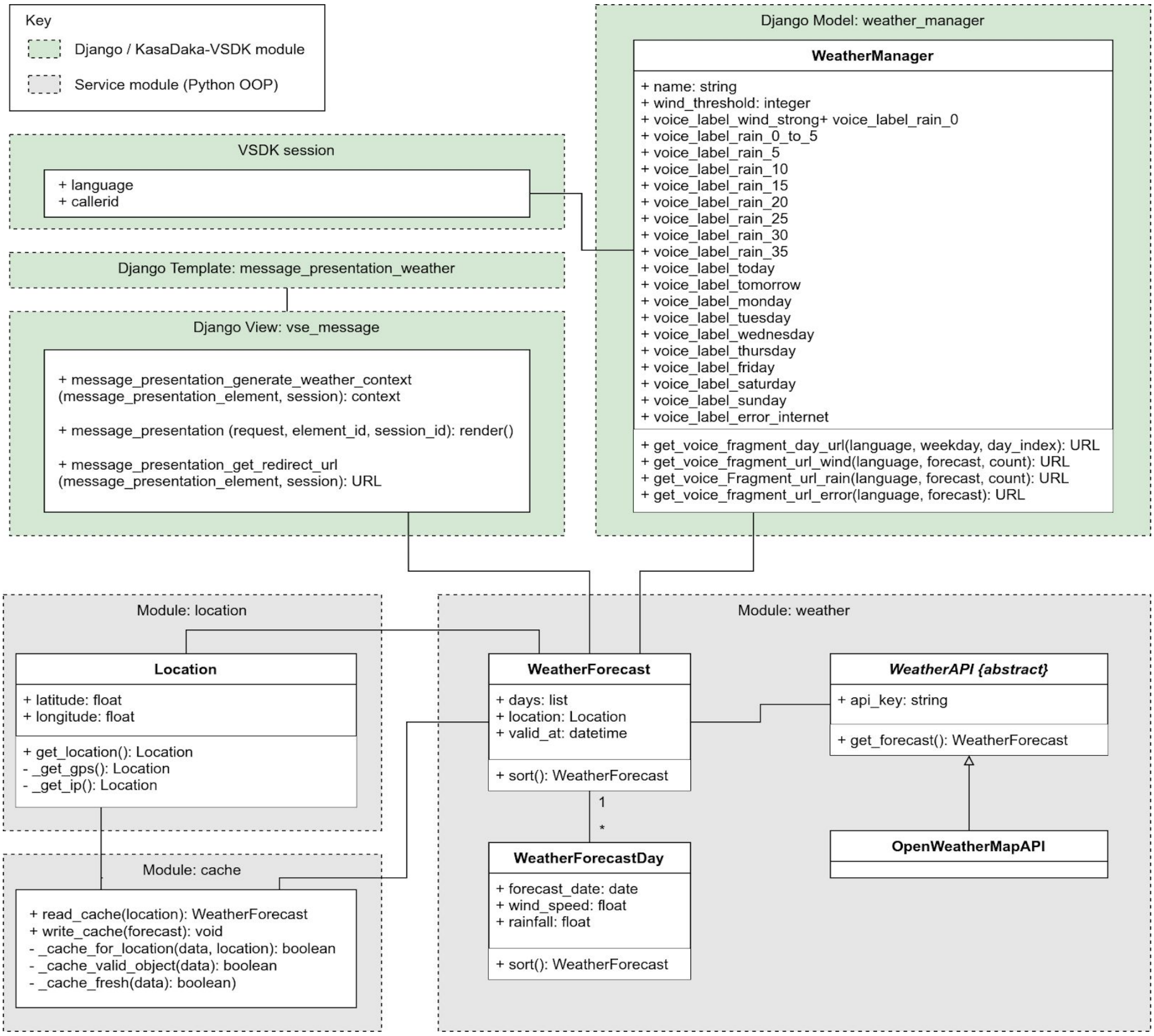

Figure 7: UML class diagram of data model

Implementation Results: Due to system location during development (local or Heroku datacenter), a location in Ghana was mocked, although a GPS software module is deployed as a stand-in on the backend. The main challenge while developing was finding a suitable model for geo-location since telephone calls from relevant users do not transmit caller location information. Therefore, without a GPS module added on to the KasaDaka, a fallback mechanism during development time was

implemented by reverting to public IP geo-location, obtained by making HTTP calls to a freely available web service[1]. Its drawbacks include less accuracy and addition of a dependency to the system. Certain VSDK modules were adapted in order to implement a dynamic day-bound behavior of the logic of the forecast voice selection, based on the weather APIs response. This preventive choice helped avoid a more intricate and hard to debug chain of static contents.
To improve performance, a caching mechanism for forecast data was added in order to prevent making repetitive calls to the weather API and stay within its free tier. If a caller requests a forecast that has been created since the last five hours for the same location, the cached forecast will be returned. An optional background job (triggered via a cron job) to update the cache in regular intervals helps further in environments of intermittent Internet connectivity. To ensure robustness, network error handling was implemented and the user gets informed in case the service cannot retrieve weather data or access local cache. All modules extending the existing VSDK with weather-specific logic employ unit tests. Additionally, two integration tests were developed to test with and without Internet connection. Both tests verify the call flow starting from the starting voice service API endpoint until the terminating responses.

14. Pointer to the VXML code
https://github.com/contervis/KasaDaka-VSDK

15. Pointer to how to access the application
http://infinite-waters-84590.herokuapp.com/vxml/start/2
Call the number 020-3697664 associated with the application (Switcher) or view the short demonstration: https://www.youtube.com/watch?v=8tB0jlxmyZQ**.** A detailed installation guide is also available in the code repository (linked in Section 13).

16. Short Usage Scenario
Upon calling the number, select a language (currently English or French) for preferred interaction. Choose between a five day forecast or a mid-week forecast to hear real-time rain predictions and storm alerts.

17. Feedback questions (attached in the Appendix)

18. Discussion of Scope and Fidelity
Scope: Based on use case analysis, background research and technological limitations, it was decided that the application would have language selection to report rainfall and wind forecast for the current day and upcoming five days. This essential requirement of the application was fulfilled. English and French were selected as initial language options. However, provisions have been made for easily extending the application through addition of new languages by providing detailed documentation. Due to the absence of GPS hardware, the user location functionality could not be directly tested. Regardless, the application tries to first retrieve GPS coordinates else falls back to retrieving the public IP address to obtain a more generalised location. Design iterations were focused on improving user experience through the addition of several features. First, to improve message content, wind information would be presented only in case of a storm. Next, users could additionally choose to hear predictions for a specific day of the week or midweek predictions. Finally, an option to replay forecasts would be added. Rather than have a general storm warning, the application wished to classify storms as dust or thunderstorms. For application performance and robustness, a local cache and error handling mechanisms were required. The system in its final state could not implement forecasts based on the day of the week due to implementation restrictions. The forecasts could be played from the start until the end of the week and from the middle of the week until the end. It could not however be played for specific intervals or days of the week due to HCI considerations. The

[1] https://www.geojs.io

option to replay messages was not added due to time constraints. Also, it was also decided that the type of storm would not be addressed by the voice message. Additional research into weather conditions would be required to accurately determine the factors that constitute a dust or thunderstorm for a region. The technical implementation would only be a set of conditional checks to classify storms. All other design requirements and error handling mechanisms were fully met.
Fidelity: The data for a location is obtained from OpenWeatherMap (OWM). The quality of the forecast relies on the data gathered by this public API. Being a public API, it is hard to ensure forecast accuracy and geographical coverage that might not be akin to the more sophisticated meteorological agencies. It was noticed that forecast data was accurate only to a certain geographical extent. Strong wind alerts were not generated based on regional examination of weather conditions but were more generalised. This could be a problem in a region like West Africa that comprises varied geographical features. A wind of 30 kmph might lead to a storm in desert-like regions of Mali while not be considered threatening in a more urban country like Ghana. A local cache is employed to enable the application to store forecasts and access it in case of losing network connectivity. This cache gets updated based on a request to fetch new data if it has been more than five hours since the last update. A more efficient way is to install an optional background job that regularly updates the cache without caller intervention. At present, the application also does not account for load balancing in case of multiple calls being made at the same time.

19. Conclusion

Global warming has resulted in significant changes to weather patterns, especially rainfall [8]. This is a serious problem for farmers in the Sahel. For years, farmers have based their farming practices by observing their environment around them and using traditional weather knowledge [8]. Because of unpredictable weather, they are no longer able to rely on their traditional practices and make timely decisions for their crops. Additionally, there is a lack of a systematic manner through which they can alternatively gain rainfall information. While attempts have been made to introduce modern technology, there is also a certain reluctance to use and accept them. Using voice technology, this service attempts to enable farmers in the region by providing them with concise and relevant rain and storm information based on their approximate geographical location. The service focuses on user accessibility by designing interaction and message content around user requirements and local context. The service is designed such that it can be accessed with very minimal requirements from the user. This, combined with minimal user interactions, the goal remains to ensure that the service can be better accepted by those reluctant to use technology.

As a future extension, the service proposes sending automated voice based storm alerts to users. The service additionally proposes the use of crowdsourcing to gather feedback from users regarding quality of predictions. This feedback can be integrated into the system as a whole and shared with the weather services to determine areas lacking in terms of forecasting.

Bibliography

[1] Bon, A., deBoer, V., & Dittoh, F. (n.d.). W4RA Use Cases/Services Created For/With Farmers and Radio Stations in Ghana, Burkina Faso, Mali and DRC.
[2] (n.d.). Retrieved from https://www.kasadaka.com/index.html
[3] Aker, J. C., & Mbiti, I. M. (2010). Mobile phones and economic development in Africa. Journal of economic Perspectives, 24(3), 207-32.
[4] Network Coverage Maps. (n.d.). Retrieved from https://www.gsma.com/coverage
[5] Nuttall, B., Ben, Laloux, D., & Ellen. (2014, April 2). More on Raspberry Pi in Africa. Retrieved from https://www.raspberrypi.org/blog/more-on-raspberry-pi-in-africa/
[6] Ho, Melissa R., et al. "Human-computer interaction for development: The past, present, and future." Information Technologies & International Development 5.4 (2009): pp-1.

[7] Cohen, Michael A., Todd S. Horowitz, and Jeremy M. Wolfe. "Auditory recognition memory is inferior to visual recognition memory." Proceedings of the National Academy of Sciences 106.14 (2009): 6008-6010.
[8] Littmann, Thomas. "Rainfall, temperature and dust storm anomalies in the African Sahel." Geographical Journal (1991): 136-160.
[9] Van Vliet, H., Van Vliet, H., & Van Vliet, J. C. (2008). Software engineering: principles and practice (Vol. 13). Hoboken, NJ: John Wiley & Sons.
[10] Young, Anthony. Agroforestry for soil management. No. Ed. 2. CAB international, 1997.
[11] Lago, P. (2019, May). Architecture design decision maps for software sustainability. In 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS) (pp. 61-64). IEEE.
[12] Isakowitz, T., Bieber, M., & Vitali, F. (1998). Web information systems. Communications of the ACM, 41(7), 78-80.
[13] Gurman, A., Rubin, S., Roess, A. (2012). Effectiveness of Health Behavior Change Communication Interventions in Developing Countries: A Systematic Review of the Literature. Journal of Health Communication, 82-104.
[14] Park, J. J., Arabnia, H., Chang, H. B., & Shon, T. (2011). IT Convergence and Services. Springer.

Appendix

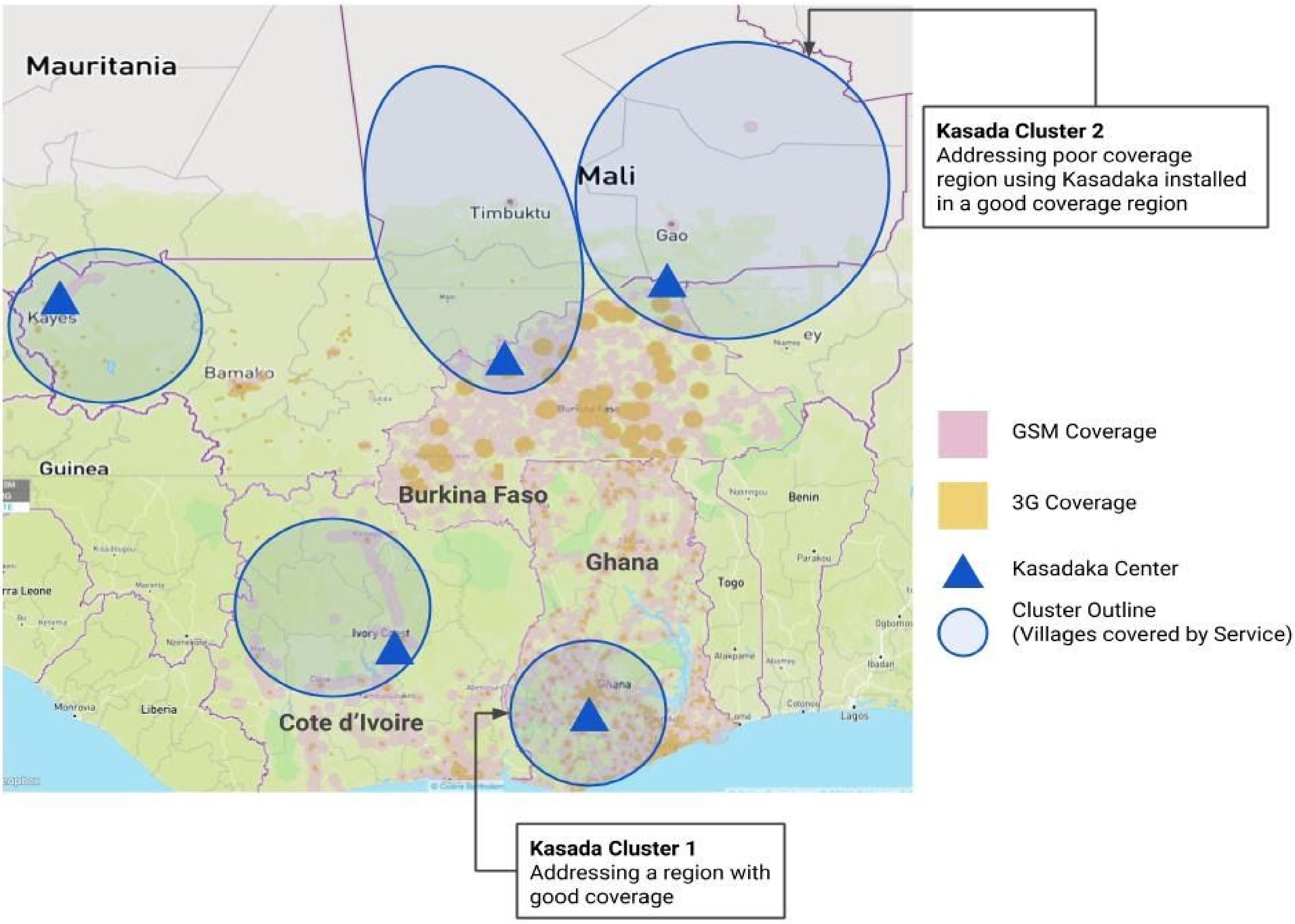

Figure 8: Map of Western Africa showing how the service proposes using clusters to identify groups of villages and assign a KasaDaka to them. A low coverage area (Mali) and a high coverage area (Ghana) are highlighted as an example.

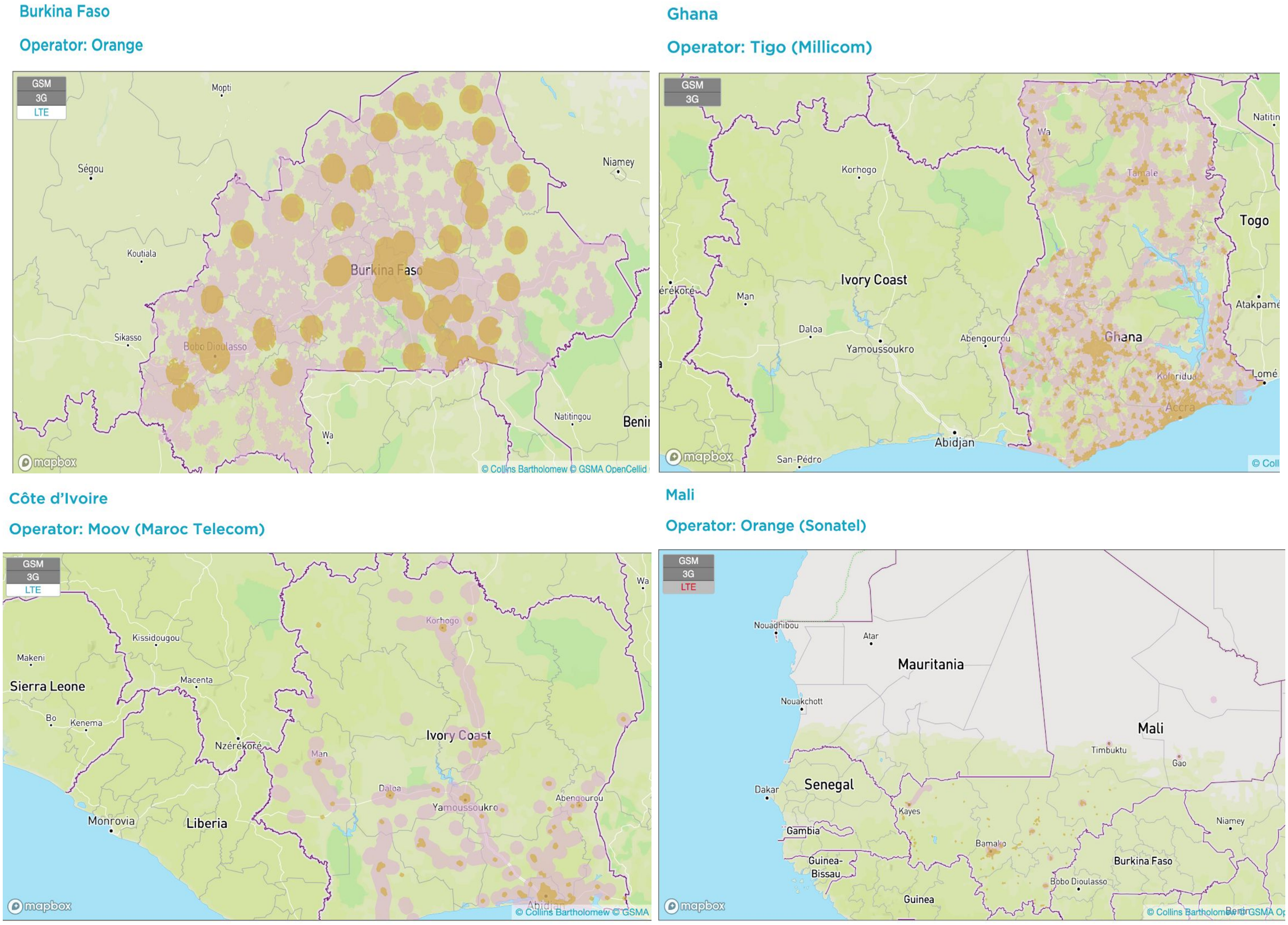

Figure 9: GSM (purple) and 3G (yellow) Coverage Maps for Burkina Faso, Ghana, Cote d'Ivoire and Mali [4]

Feedback and Questions

Presentation feedback

Question 1.0: "The way geolocation is implemented (by using the GPS location of the KasaDaka) does imply that a physical unit with a unique phone number has to be deployed inside the same region where the user requesting the forecast is (to provide a decently accurate report). This would introduce problems in rural areas which are too remote for (affordable) internet access. Have you thought about adding location options to the request from the user (farmer in this case)? Would SMS not be a better fit there? Since for a phone call the user would need to answer the phone at that exact time"

Feedback 1.1: Integrating SMS is an interesting option. In Mali, the feedback is that most people do not use SMS. Also in N-Ghana, many people will not be able to effectively use sms

Response: *SMS would indeed be better but we are concerned that the users we are targeting do not really use SMS as per Victor. We are wondering if we can leave a voice message on their phone instead. The farmers do not need internet access. All they need is a mobile network connection that they can use to dial in our service's number. The KasaDaka platform itself has a 3G GSM dongle that will be connected to the internet to rainfall data.*

Question 2.0: "It is a bit unclear to me whether the farmer is calling just once to subscribe to the service and then gets alerts until it unsubscribes, or is it just one time thing where they call, and then subsequently receive an alert, and need to call every time to receive alert. Could you clarify this for us?"

Response: *The user simply calls the service each time they want to hear rainfall/storm information. We do not intend to have any user database since we want to make the application as simple and user friendly as possible to cater to the users who are mostly hesitant to use technology.*

Question 3.0: Since your application targets farmers specifically, have you thought about giving some more information that is related to them? Such as harvesting tips for certain crops or the like.

Response: *At this time, we have kept the project's focus very specific to address the lack of timely and accurate rain and storm alerts. But definitely, the suggestion is an excellent further improvement!*

Assignment 2 feedback on Canvas

Feedback: "Simplistic interface, which is good. Possibly too much info at a time; consider selection of forecast timeframe"

Response / implementation: *We now provide only wind information in case of storm; functionality to select a time-frame (skipping forecast for today and tomorrow). Always the possibility to terminate the call earlier.*