# Unpacking Grant Requirement

**Describe high-capacity informatics infrastructure with the following features:**

* **high-security-compliant**  *(doing thing securely/HIPAA compliance/careful with HPI)*
* **data integration** *(able to handle complex data inputs from various sources)*
* **data storage** *(has secure/scalable/performant storage)*
* **data management** *(able to clean, transform and analyze)*
* **data sharing** *(data interoperability and efficient mechanisms for distribution of data or results)*

Above, grant requirement text is in bold; each bullet point has annotations in italics to give a more functional definition of each feature. Overall, I think the features individually sound pretty familiar and things we have experience doing at Dornsife. However, the key underlined word ‘informatics’ is not a term which is commonly used at Dornsife.

Informatics studies the interplay between people and technology (specifically computer science and computing hardware). For example, this interplay could take place between statisticians and technology; we use computers to store data (as opposed to paper records), to do computations (opposed to calculating by hand), to share findings (instead of draw plots).Within this context, when the funding committee is asking use to describe high-capacity informatics infrastructure what funding committee really means is: If this grant is funded, you will need to use computer technology to carry out your proposed grant objectives ... what is your plan to use computing safely, effectively, and efficiently?

A thorough answer to this question isn’t a simple we will use technology X for this and this but rather communicating our informatics core values and propose some ways to implement them.

# Core Values

At the Urban Health Collaborative, many of our research efforts involve inter-institutional data collection then downstream research synthesis. A concrete example is a UHC led research project that examined COVID-19 inequities across 30 major cities, we found that local health departments were gathering and reporting through ad-hoc methods and there was a lack of a coherent inter-institutional approach towards COVID-19 data. However even at a project level, this challenge can exist. Additionally, SALURBAL is a research initiative at the UHC which focuses on global health through the collation and harmonization of multi-domain data across 11 Latin American countries. Towards the latter stages of the project funding cycle, one of the goals is to develop a data-platform to systemically make project data accessible; during this effort we discovered that there were inter-working group data interoperability issues which prevented efficient access to the data at an institutional level which we had to address post-hoc.

In both these cases lack of data interoperability was a bottleneck for the rapid synthesis research and policy recommendations. These issues were a wakeup call to our institution that success of large-scale research projects requires not only sound statistics/methodology/policy-engagement but also the development of a well thought out informatics infrastructure to prevent data bottlenecks. The core values of informatics at the UHC include: 1) data-interoperability 2) informatics competencies 3) web-first accessibility. As discussed above, there is critical research value in data availability, and we are committed to designing infrastructure which will maintain data-interoperability as the project evolves in complexity; this involves allocation of funds to informatics-focused personnel and to engage them during project inception in regard to implementation of FAIR data principles. Secondly, as a project scales, multiple working groups may emerge. To maintain data-interoperability standards across working groups it is key to have mechanisms in place that develop informatics competencies; this involves both on-boarding training for new faculty or staff that introduces them to core informatics concepts, standards and skills related to the project as well as a centralized informatics core to aid during project growth.

Lastly, we have found that thinking web-first thinking – prioritize having an accessible web application that documents project progress from data extraction, transformation, database loading to research synthesis- is a practical way to enforce FAIR data principles and maintain data interoperability through out the project life cycle. Developing custom web tools as a part of project forces us to not only explicitly codify FAIR data principles but also allows us to systematically evaluate and enforce these standards onto our data. By developing user friendly and secure machine-human interfaces for internal project use, we can automate a lot of the informatics housing keeping tasks such as quality control, data integration from multiple groups, or data distribution without giving direct access to secure databases; this relieves staff from the burden of redundant tasks that are prone to human error. In addition to automation of informatic inefficiencies, this approach when adapted early on for internal use also builds the foundation for a public facing version when the project is mature enough.

# Technical Recommendations

While the core values and intentions above are importantly, equally important is what tools we use to implement them.

## Hardware

From a hardware perspective, commercially available cloud infrastructure such as Azure have some key features that are particularly attractive for this project including:

* **vertically integrated** service portfolios that include version control, storage, computing, and application hosting
* **scalable** to enterprise level data allows us to trust that the infrastructure we start with at day 1 will be able to handle what ever data requirements we may face further down in the project
* **cloud based** means we do not have to invest in hardware or personnel to maintain servers
* **secure:** core services can be configured to be HIPAA compliant and store HPI

The three largest main cloud infrastructure platforms Microsoft Azure, Google Cloud and Amazon Web Services are largely at parity when it comes to breadth of services, price, and performance. Comparing the three at a high level is largely consequential; like comparing Kellogs and General Mills. As all three are developed for enterprise-grade development we can trust them to fulfill our data requirements. However, when taking into consideration existing infrastructure and experience at Drexel, Azure is the clear winner. There is already Azure infrastructure in place at Drexel where Central IT has set up an Azure account which is used by the UHC. For example, the SALURBAL project utilizes Azure Blob storage for its database, Azure Static Web App services to host its data platform and Azure Function Apps to do serverless computing for its front-end applications. Importantly, the West Philadelphia Promise Neighborhood project has used Azure storage for HPI data.

## Software

The interplay between humans and hardware really occurs through software. There are two main principles which guide how we use software.

**Version-control**. Analytics is code. Maintaining production grade code is key to reproducible project development. We aim to integrate version-control management through using Git and an online Git platform such as GitHub to future proof our analytic codebase from various potential problems including collaborative coding amongst a remote work force, potential staff transitions, automation of continuous integration of code changes into a single codebase or continuous delivery of updates to web applications.

**Highly functional but minimal technology stacks**. Overengineering a solution may often times prove to be more trouble and costly than the solution itself. At the UHC we evaluate technologies based on not only functionality and performance but also on speed of development and maintainability. For examples when it comes to building a data platform, one options is the traditional CKAN pipeline which requires several virtual machines (costly infrastructure) and complex development (requires expertise in Python, SQL, JavaScript, Docker, Kubernetes, Linux). An alternative is solution is to utilize R (a skill prevalent in public health) to generate, QC and store data in Azure blob storage, build a static web application with a feature rich UI with React (JavaScript) and to handle computations on a serverless Azure Functions App (JavaScript). The latter approach not only reduces the breadth of technical expertise required but also doesn’t utilize costly infrastructure such as virtual machines to provide a performant final product.