IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) | 978-1-6654-0926-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/INFOCOMWKSHPS54753.2022.9798061

The AERPAW Experiment Workflow - Considerations for Designing Usage Models for a Computing-supported Physical Research Platform

Magreth Mushi[a], Harshvardhan P. Joshi[a], Rudra Dutta[a],
Ismail Guvenc[b], Mihail L. Sichitiu[b], Brian Floyd[b], Thomas Zajkowski[c]
[a]*Dept. of Computer Science*,[b]*Dept. of Electrical and Computer Engineering*,
[c]*Institute for Transportation Research and Education, Aviation*
North Carolina State University
Raleigh, NC, USA
{mjmushi, hpjoshi, rdutta, iguvenc, mlsichit, bafloyd, tjzajkow}@ncsu.edu

***Abstract*—The AERPAW project is an ambitious project, funded by the PAWR program of the US NSF, to create a remote accessible research platform for a research facility with some distinct features that makes its usage model unique, and non-obvious to many researchers desirous of making use of this platform. AERPAW is primarily a physical resource (not a computing or cyber-resource) - the RF enviroment, and the airspace. Experimenters can explore them through radio transceivers and Unmanned Aerial Vehicles, both under the Experimenter's programmatic control. Since the entire workflow of the user is through the mediation of virtual computing environments, users often tend to think of AERPAW as a computing resource, and find some of the experiment workflow counter-intuitive. In this paper, we articulate the challenges and considerations of designing an experiment workflow that balances the need for guaranteeing safe testbed operation, and providing flexible programmatic access to this unique resource.**

***Index Terms*—Wireless, testbed, 5G, NextG, UAV, UAS, drone**

I. Introduction

The wireless era is new in the history of the modern world, with unprecedented bandwidths and ubiquitous mobility unleashing new applications and capabilities. Revolution of so many things connected wirelessly in a way never done before allows technologies such as driverless cars, virtual/augmented reality, drones, and massive Internet of things (IoT) moving into reality. The fifth generation (5G) cellular networks will offer unprecedented data rates, ultra-reliable low-latency communications, and massive machine type connectivity that will allow enterprises and specific use cases to flourish in ways we now cannot predict. Central to the vision of 5G is the availability of immense bandwidths that will enable, for the first time ever, cognition and video control of robots, and moving platforms, particularly in areas where the dense 5G wireless network exists.

Major new use cases for advanced wireless technologies, once thought to be science fiction, are emerging in the unmanned aerial systems (UAS) spaces [1]–[3]. Major package delivery companies such as Amazon and UPS [4], [5] have invested heavily in UAS capabilities, and many other military, government, and consumer applications of UAS have attracted major attention, e.g. by Google [6], Uber [7], Boeing [8], Zipline [9], Flytrex [10], and Matternet [11], among others. The U.S. wireless industry will provide the infrastructure and capabilities to service this revolutionary move to UAS and flying vehicles. While various cellular providers (AT&T [12], Vodafone [13]), vendors (Ericsson [14]), and chip manufacturers (Qualcomm [15]) have recently been studying cellular connectivity for aerial links, yet very little is known to date about how to properly design, develop, analyze, and test this major leap in the world's infrastructure of UAS.

The Platforms for Advanced Wireless Research (PAWR) funding program of the US National Science Foundation (NSF), with funding provided both by NSF and an industry consortium engaging public and private partners, has the goal of establishing several large scale advanced wireless testbeds enabling experimental research at scale. The PAWR-funded testbeds will be open to both academic and industry researchers, enabling unprecedented access to advanced wireless technologies. Participating companies provide in-kind investment and receive in return priority use of the corresponding testbeds, potentially presenting research experimentation opportunities not available elsewhere.

The Aerial Experimentation and Research Platform for Advanced Wireless (AERPAW) is the third such testbed funded under the PAWR initiative. The project started in September, 2019, and the AERPAW facility became generally available, with a minimal set of experimental resources, for the first time in November, 2021. Over the next two years, the set of experimental resources is expected to increase.

The unique nature of AERPAW poses both an opportunity and a challenge for the researcher interested in designing and executing experiments on AERPAW. Experience so far indicates that the initial challenge of understanding the programming orientation of the AERPAW facility, and starting to make use of the testbed, may be a crucial one for many potential users. In this paper, we present AERPAW's usage models and experiment workflows, as well as the considera-

978-1-6654-0926-1/22/$31.00 ©2022 IEEE

tions and constraints that dictated them, in the hope that this will help new AERPAW users gain insight into the logic of these workflows, and adopt AERPAW for their experimental needs with ease.

II. AERPAW Facility

AERPAW was envisioned to fill the need of enabling meso-scale UAS experiments in a production-like wireless environment, and serve as both an advanced wireless research facility, and a UAS research facility; but most importantly, provide a testbed for research that jointly addresses both.

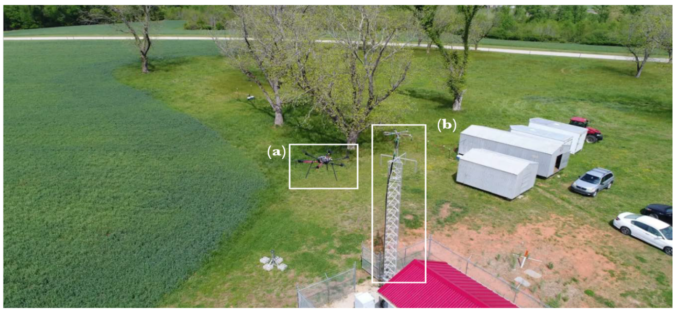

Fig. 1: AERPAW experiment in progress at Lake Wheeler: (a) one UAV mounted portable node and (b) one tower-mounted fixed node is visible

Physically, the testbed is hosted at sites in and around the NC State University campus in Raleigh, NC, USA. Central to AERPAW's unique characteristic is the availability of Unmanned Aerial Vehicles (UAVs) in the testbed that can be placed under the direct programmatic control (of trajectories) of the researcher. In conjunction with the programmable USRPs also available for direct programming by the researchers, as well as other real-world, commercial radio equipment, this provides the NextG wireless researcher a facility for research experiments not practicable in any other facility at this time.

At a very high level, the facility includes a number of *Tower* locations, at each of which some combination of AERPAW programmable SDRs and commercial radio equipment is permanently installed. The SDRs are controlled by servers, or companion computers, installed in each location that represent edge-computing capabilities. These fixed node locations are distributed over the extensive Lake Wheeler Agricultural Fields of the NC State University, and some nodes are also installed in the more conventional Centennial campus of NC State. In future, some nodes may be installed in adjoining city and town areas. The complement of these Fixed Nodes are AERPAW's Portable Nodes, also consisting of a computer and SDRs, but smaller ones so that an AERPAW Portable Node can be mounted on a UAV. The computer on a Portable Node also controls the UAV itself.

To provide an overall idea of the testbed expanse, Fig. 2 shows the location of currently active (red) fixed nodes as well as those to be commissioned in the near future (yellow). To provide a sense of scale, the north-south extent of the area shown is just over three-and-half miles. The bottom left cluster of yellow nodes occupy a central position in the Mid-Pines

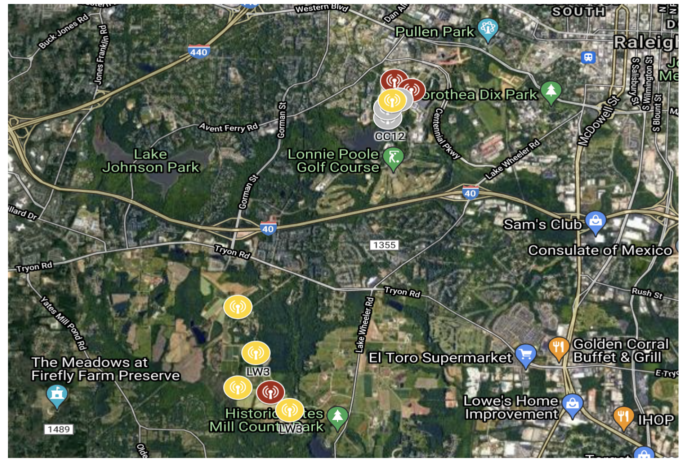

Fig. 2: AERPAW initial deployment area

Lake Wheeler Field Laboratory of NC State, where further buildout is planned for future.

To be able to conduct such experiments, while remaining in compliance with the stringent regulations of the Federal Communications Commission (FCC) and the Federal Aviation Authority (FAA) for SDR and UAV operations respectively, requires appropriate certifications and exemptions. The AERPAW platform has already acquired the relevant ones, and extending to further exemptions is on our roadmap; these certifications and the experimental capabilities unleashed by them (and the accompanying practical experience) are critical resources that AERPAW brings to its experimenters, beyond the physical resources. The details of the certifications and exemptions are available in the AERPAW User Manual [16].

Extensive information about the facility, equipment details, software support, etc. are provided in our public online User Manual [16], the AERPAW website [17], and representative research experiments as well as design details of the platform have been described in our prior publications such as [18], [19]. We do not repeat any of that content here, beyond the above preliminaries. Instead, we next describe some aspects of AERPAW architecture, to facilitate understanding of the experiment workflow that follows.

III. Usage Model – Design Goals and Constraints

As is clear from the above, **AERPAW is primarily and essentially a testbed of physical resources, not computing resources.** The crucial part of these physical resources are (i) the RF environment and the airspace that the AERPAW operating areas represent, (ii) the physical equipment (SDRs and UAVs) that AERPAW provides to leverage those environments for experimental studies, and (iii) the expertise (and consequent exemptions) in conducting such studies in compliance with FCC and FAA regulations that AERPAW represents.

A. Deep Access

As an NSF-funded testbed, AERPAW has the goal, and the charge, to serve a broad range of researchers nationwide, rather than just a small set of specific domain researchers with deep-but-narrow portfolio of domain expertise and interest. In other words, we do not aspire only to be of use to the few researchers, typically in industry, who are already well-versed in SDR, 4G/5G and UAS programming, and are simply seeking a physical arena for their scale experiments. Rather, we also aspire to bring the capability of engaging in such research to a larger variety of researchers who may not already possess such expertise, including those in academia, with a broader range of interests – e.g. in the theory and analysis of channel models or modulation schemes, in algorithmic trajectory control and optimization, in edge-computing for real-time image analysis or cyberphysical systems, and many others.

Further, we wish to provide researchers with complete control of the equipment we operationalize in the testbed, without imposing any additional artificial barriers, or limitations of planned experimental activity. In other words, we aim to provide the researcher with "root access" to all our equipment. This is not to say that arbitrary experimental activity can be *executed* on the testbed without limitation; naturally, unsafe operation (such as RF operation outside the allowed profile of frequency and transmission strength, or UAV operation outside the allowed profile of speed, angle of attack, etc.) must be prevented from actually occurring. We describe this more specifically below, and relevant discussion may also be found in [18]. Rather, we make the choice to try to preclude such unsafe or illegal operation at actual execution time, rather than by imposing *a priori* limitations on the experiment design and planning process, or tools used for that purpose (to ensure that no "dangerous" experiments can even be defined). The reason for doing so is that the latter choice, though easier to implement for AERPAW, would take the nature of blanket bans, and preclude many interesting and legal experiments as well as actually illegal ones. For example, consider the researcher who writes a custom algorithm for dynamically selecting the transmission frequency or power, or UAV trajecory (or both) during execution. Depending on the nature of the algorithm, it might well be impossible to predict with certainty (even for the researcher himself/herself) whether the operation will stay within legal bounds during actual execution. The only way to preclude dangerous operation with certainty would be to disable programmable real-time computed control of the SDRs or UAVs altogether (and insist instead on explicitly pre-determined and pre-programmed transmission and flight), thus precluding many interesting algorithmic experiments. Finally, in future, we aspire to provide researchers with similar deep control, within practical feasibility, of real-world commercial radio equipment (such as commercial cellular base stations).

B. Regulatory Compliance Issues

Since the AERPAW testbed is physically located in the particular space it occupies, to physically access it researchers would have to travel to it, which is impractical. Therefore we must make it possible to use this testbed remotely. This, and the goal of providing full control to the researcher, requires that every capability of AERPAW equipment must be possible to access by the mediation of some computing device – most typically, commodity computers running a general purpose OS (usually Linux), but also special-purpose software to enable control of radio or UAV equipment. The remote experimenter can optionally choose to write their own custom software extensions or replacements for any part of the pre-installed software.

This, incidentally, may potentially reinforce the *incorrect* impression that AERPAW is fundamentally a computing, or cloud-computing, facility, with some peripherals accessible through the remotely accessible computers. However, as we have described above, the appropriate way to think of AERPAW is as a physical facility, mediated by computing for the dual purposes of remote access, and deep programmability.

The second issue raised by this set of goals and constraints is that of liability. AERPAW possesses appropriate exemptions and permits, as described above, to conduct a range of RF and UAV operations, and can execute such experiments legally, these permits do not automatically translate to allowing other researchers to conduct the same operations. Thus, a researcher who desires to perform an experiment on AERPAW needs to first obtain the same exemptions, certifications, and permits as AERPAW, which is obviously untenable in terms of using AERPAW as a testbed. It would also not be practical for AERPAW to verify such permits. Further, if unsafe operation occurred, it would be unclear where the liability should lie – with AERPAW, or the experimenter.

C. Batch-mode Access

To reflect the main purpose outlined above, balancing the needs to afford deep programmability to the researcher using AERPAW, and to ensure safety and regulatory compliance, we reached the conclusion that the canonical usage model of AERPAW must be that of *batch-mode access*. In the computing context, this feels like an archaic model; however, it is a standard model for facilities that are fundamentally physical in nature, such as large telescopes or microscopes. Batch-mode operation is thus a natural fit for an essentially physical facility such as AERPAW.

However, the AERPAW canonical model is somewhat more ambitious than the standard batch-processing model for large physical resources. Most such models not only allow but actually require the user to not only produce written documents to describe the intended experimental activity, but also to write executable scripts that actually drive the physical device(s). The same is true of AERPAW. However, our goal go significantly further, in two ways. First, for canonical experiments, we hope to eliminate the step of documentary or any other human-in-the-loop interaction altogether, and allow the AERPAW experimenter to define an experiment and stage it for batch execution completely programmatically. At the time of this writing, some human-in-the-loop operation is still required, although transparent to the user; but we expect to eliminate it soon with ongoing development of the platform. Note that custom experiments that do not fit the automated workflow can always be imagined, and such experiments will always require human-mediated planning; see below for more details. Also note that even for canonical experiments, human action is required for the actual execution of the experiment, since UAVs have to be staged, tested for safety, etc.; what is automated is the process of experiment planning and batch submission; again see below. Secondly, we aim to provide the researcher with a single experiment preparation methodology that will enable them to write the minimal scripts for using AERPAW for oft-repeated experiments, but also to use the same environment to develop sophisticated custom software extensions of their own. For example, a researcher may want to encode their own experimental RF waveform for specialized use such as air-to-ground communications.

D. Staged Preparation Discipline

To enable this vision, a complementary virtual and physical access discipline was envisioned by the AERPAW architects. During experiment preparation, researchers using AERPAW are provided access (with root permissions) to a number of virtual computing nodes, one corresponding to each physical AERPAW computing node. From the user's vantage point, each node is identical to the corresponding testbed node; in fact, they are identical in software capabilities, but do not have physical resources such as SDRs or UAVs (or other commercial radio equipment) connected to them. However, the software behaves the same general way in this environment as in the actual testbed environment, thanks to a custom emulation environment, described more fully in [18]. This not only enables the experimenter to develop their software with ease, running it in the emulated environment through the development cycle, but also makes for a seamless transfer of

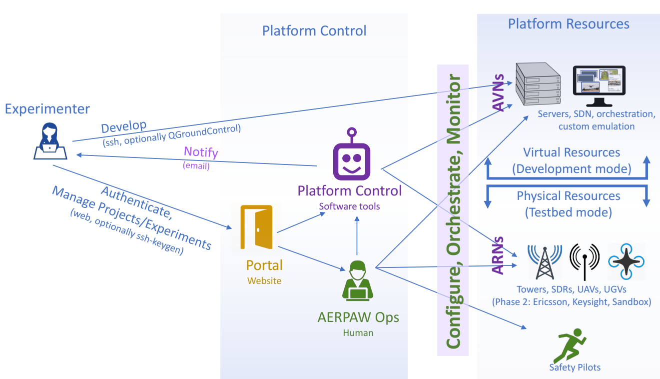

Fig. 3: AERPAW Entity Relationships

Fig. 3 shows the entity relationships of the platform at a very high level, that captures the above.

In summary, then, **AERPAW is a batch-mode facility**. Experimenters develop experiments in a virtual computing environment, and submit experiments for execution on the physical testbed once development is complete. AERPAW Operations personnel (Ops) then execute these submitted experiments in the physical testbed environment, and collect the output of the experiments as designed by the Experimenters, which are available for Experimenters to view and analyze back in the virtual environment.

This is not an arbitrarily decided constraint, but a considered architectural choice as we have discussed above. In operating a facility with programmable radios and programmable air vehicles, we are obligated to make, and uphold, certain guarantees to the FCC and FAA. However, we also want to allow Experimenters the ability to program those radios and air vehicles, ideally without needing to become fully conversant with FCC and FAA regulation details, obtaining exemptions, or expertise at techniques for ensure compliance.

Batch mode operation allows us to interpose critical filters and monitors into the Experiment code execution flow that allow us to guarantee safe and compliant operation. It is one of the most valuable features of the AERPAW platform that we assume this guarantee ourselves, rather than passing on the responsibility for compliant operations (and liability for non-compliance) to the Experimenter.

IV. AERPAW Architectural Approach

While this paper focuses entirely on platform usage model and experimenter's point-of-view, some nomenclature and concepts related to the AERPAW architecture are required as preliminaries, which we present in this section.

A. AERPAW Node Classes

AERPAW Compute Node: A computer, real/virtual, that contains client-side AERPAW control software. We often refer to this as a Companion Computer, since it accompanies whatever experimental resource (radios, vehicles) an experimenter is interested in experimenting. We also sometimes refer to this simply as an **AERPAW Node**. An AERPAW Node that is instantiated physically for an experiment is an AERPAW Hardware Node (AHN); a VM or container that is instantiated to be a close digital twin of an AERPAW Hardware Node, for the purpose of experiment preparation, is an AERPAW Virtual Node (AVN).

AERPAW Cloud Node (ACN): This is a special case of an AERPAW Node to which no other controllable hardware resources are connected, and therefore for which the physical location is not important. For the experimenter, it represents a pure computational resource. Such a node would be instantiated as a virtual computing entity (VM or container) in the AERPAW central datacenter (DCS).

AERPAW Hardware Radio Node (AHRN): An AHN to which one or more experimental wireless hardware resources are connected. An AERPAW Radio Node implies specific physical instantiation at some location at which the radio resources exist, and is not a virtual resource (although it may contain multiple VMs or other execution contexts within itself). We distinguish between **Fixed** and **Portable**, called **AFHRN** and **APHRN** respectively. Each of these two types of AERPAW Radio Nodes can, of course, have a VM instantiation (for development or emulation), in which the experimental wireless hardware is replaced with an emulation, thus providing AERPAW Fixed Virtual Radio Nodes (AFVRNs) and AERPAW Portable Virtual Radio Nodes (APVRNs).

A Fixed Radio Node (AFHRN) is installed at some specific location, (semi-)permanently, and can only be used in-situ by experimenters. Essentially these nodes are "always-there" or "persistent." They may be powered down for intervals by testbed control but are "always-there" to be remotely powered back on.

A Portable Radio Node (APHRN) is compact and self-powered, so that it may be moved during an experimental session. As such, it is only guaranteed to be available to testbed control at specific times, thus "ephemeral." It may appear at various times with various L2 and L3 addresses when it does, but needs to be recognized by testbed control as the "same node."

Portable AERPAW Radio Nodes can move during an experiment, but do not necessarily do so. We speak of the role played by a portable node as "Stationary" or "Mobile," depending on whether it is required that the node should move during an experiment. The same Portable Node may be stationary in one experiment and mobile in a different one. Fixed nodes, of course, can only play a "Stationary" role in any experiment.

There are many reasons why it might be desirable to use a Portable node in a Stationary role for an experiment. It might be desired by an experimenter to position a Stationary node where no Fixed node has been installed. A special case of this is when a node is desired to be positioned in an aerial location, but not desired to be mobile during the experiment. A passive vehicle such as a tethered balloon or helikite may be used to position a Portable node in a Stationary role, for a long-term experiment.

Finally, some Fixed and Portable Hardware Nodes may be placed in the controlled Sandbox facility where the connected experimental wireless hardware has been modified to allow dense indoor placement. This creates AERPAW Sandbox Fixed Hardware Radio Nodes (ASFHRNs) and AERPAW Sandbox Portable Hardware Radio Nodes (ASPHRNs).

B. Entities Associated with AERPAW Nodes

AERPAW Experimental Radio: Any wireless equipment to which AERPAW will provide experimenter access through a Companion Computer. A large fraction of these are expected to be SDRs, but other commodity wireless equipment such as WiFi or LoRaWAN also fall into this class. The equipment we consider in this class have open interfaces, and are deeply programmable or at least comprehensively configurable. An AERPAW Radio Node houses one or more such Experimental Radios.

Third-Party Blackbox Equipment (3PBBE): Incorporating these is one of the more ambitious aspects of the AERPAW architecture. In this category, we place any device (or a collection of devices that together form a vertical set, or ecosystem, of devices) that is essentially commercial equipment, meant for production use. A commercial hardware 4G or 5G base station is an example of this class of equipment. However, while such equipment is not intended to be used for experimental purposes, and thus typically designed without deep programmable or configurable interfaces and therefore difficult to experiment with, many experimenters would like to experiment with them nevertheless, for the realism of research results afforded by such devices.

At this time, the 3PBBE equipment integrated into the AERPAW platform consist of an Ericsson 4G/5G base station-RAN-core system, multiple Keysight RF sensors, and a system of four Facebook Terragraph radios.

Vehicle: Carrier of an APRN during an experiment during which the portable node acts as a mobile node. A vehicle may be programmable, providing experiment-controlled mobility (UAVs, Unmanned Ground Vehicles or rovers), or not, providing vehicle-controlled mobility (shuttles, SUVs, manually piloted UAS, backpacks on cyclists or pedestrians).

Virtual Computing Entities: In order to provide the experimenter with root access, but still retain supervisory override authorities (to preclude unsafe operations), we run multiple virtual computing entities in each AERPAW Node during Testbed execution. They can be instantiated as actual Virtual Machines (VMs) or containers; at this time AERPAW uses containers exclusively. In the rest of this paper, we use the terms VM and container interchangeably to indicate such virtual computing entities. AERPAW has two main virtual entities: **E-VM** and **C-VM**. The E-VM is the virtual computer afforded the experimenter, in which the experimenter enjoys root access. The experimental SDRs are directly connected to and under complete control by the E-VM, and any code running on it. Each AERPAW Node also runs a C-VM container, whose function is monitoring and override. The C-VM has its own monitoring SDR, which it uses to detect any illegal operation attempt by the E-VM's SDRs.

The vehicle is connected in tandem to both the E-VM and the C-VM; the E-VM's commands to the vehicle do not need to pass through the C-VM, but can be observed by the C-VM, which can exercise its supervisory authority to withdraw vehicle control from the E-VM, if unsafe or illegal operation of the vehicle is attempted by the E-VM.

V. Using AERPAW – Experiment Workflow

Our preparatory survey of the advanced wireless research community, as well as our ongoing communications with it, strongly indicates to us that experiments in which experimenters program the UAV trajectories, and the SDR transmissions, are the ones of greatest interest to potential AERPAW users. Accordingly, we consider this to be the standard, or "canonical" AERPAW experiment. Therefore, in the rest of this section we first concentrate on this workflow. Subsequently, we briefly discuss non-canonical experiment scenarios and workflows.

A. Canonical Workflow

The core workflow consists of providing multiple stages of experiment preparation options. These are designed with various goals in mind. Given the cutting-edge nature of some of the technology we plan to provide in the testbed, we also realistically envision that at least some of the users will not have previous programming experience with at least some of the resources that will be available to them in the testbed - this means that they will have to go from first encounter with the capabilities of such equipment to designing meaningful batch-executable experiments with them, and do it all by remote operation. This is likely to pose a challenge for at least some researchers, and the AERPAW experiment preparation process has been designed to make this as efficient and smooth as possible. In addition we provide extensive tutorial material to provide the fastest possible on-boarding and successful experimentation.

Before this workflow can be embarked on, a user must gain an authorized experimenter status in AERPAW, and login to the platform, using a web-based portal. This aspect of AERPAW is quite similar to many research platforms, and is not described in detail here. User authentication is based on a federated single-signon system.

Creating a Saved Experiment: An experiment begins when an authenticated and authorized user indicates through the web portal that they wish to create a new experiment. As part of the creation, the user must specify how many AERPAW nodes (and their corresponding role) their experiment will contain. For each Fixed Node, the user must also pick a specific location out of those available in AERPAW. For each Portable Node, the user must also specify the kind of vehicle desired if any (whether experiment-controlled or not), and a starting geolocation. For each node, unless only a single available profile image is available, the user must also select one of the available profile images. At this time, our model is that the choice of node also implies that the user intends to make use of the entire set of radio and other resources available on the node specified (none on Cloud Nodes, a specific set for each Fixed Node, a standard set for Portable Nodes). At a later time, we may enable the user to choose a subset of resources for any given node, and avoid activating (for emulation) or energizing (for sandbox and testbed) the rest during the experimentation. Upon submission of this request through the Web Portal, the nodes are created and saved. At this time, the experiment is considered to be in the "Saved" state.

Development: A user can select a "Saved" experiment in the dashboard available on the web portal, and choose to "develop" it. In response, the platform will create running VMs out of the AERPAW Compute Store, using the E-VM images previously saved, and provided secure login access to them via ssh keys or similar. The experiment is now considered in "Development" mode. The user can do basic and exploratory development on these VMs; although these VMs do not have the radio and other hardware resources that the corresponding testbed computers have, the sample profile code can be browsed and experimented with. The user can also experiment with the experiment control and synchronization software included with the profile that is required to be used to enable gated and synchronized execution of the various nodes during actual testbed execution. At any point during Development, the user can choose to save the experiment.

Sandbox: A user can select a "Saved" experiment, and request a "Sandbox" preparation environment rather than a "Development" one. The distinction is that for a sandbox execution, at least some subset of the E-VMs are loaded onto physical computers that have the same AERPAW Experimental Radios attached to them as the corresponding testbed nodes do. If an E-VM corresponds to a Portable node to be mounted on an Experiment-Controlled Vehicle, the corresponding sandbox node would also have the same vehicle attached to it. However, the sandbox nodes (and experimental radios and vehicles) are in an indoor lab facility rather than an outdoor setting, with appropriate safety modifications. The user receives a live login capability to the Sandbox VMs just as the development VMs. However, since the sandbox resources are likely to be far more limited than the compute store, this access needs to be scheduled. When the user requests sandbox access, the platform will consult the resource schedule and offer the user one or more sandboxing timeslots. When the user accepts one, that time is actually scheduled on the sandbox for that experiment, and the user receives a login to the E-VMs running on sandbox and cloud facilities similar to the development case. The Sandbox reservation also has a specific duration; the user can choose to save the experiment before that, but when the duration expires, the sandbox session is automatically terminated, and the experiment auto-saved.

Emulation: A user can submit a "Saved" experiment for "Emulation". Emulation is both a scheduled (like Sandbox) and a batch (unlike Development or Sandbox) mode of execution, and the user does not receive online access to the emulation environment. As a consequence, the scheduled time for the emulation need not be convenient for the user, and therefore the platform can schedule the emulation at the most appropriate time whenever the emulation environment is executed. Further, the user retains access to the E-VMs until the scheduled time of the emulation starting, for Development or Sandboxing, but loses control of the E-VMs once the experiment actually enters the Emulation mode. (If the experiment is not in the "Saved" state at the scheduled start time of the emulation, the emulation is canceled and must be re-scheduled.) In the Emulation mode, all the VMs of the experiment execute in a special software environment in which they interact with software or hardware emulations of the radio and airspace environment based on detailed physical models. The AERPAW Experimental Radio and Vehicles are emulated in software in the Emulation mode, no hardware is present. The complete experiment is logged exactly as it would be in the actual Testbed environment, and the log is examined after completion for a go/no-go check for safety violations, resulting in either an "Emulation-Passed" or "Emulation-Failed" flag to be saved with the experiment when it is saved back at the conclusion of the emulation. Currently we envisage a partially manual and partially automated implementation of this go/no-go check.

The user regains control of the experiment at the conclusion of emulation, but if they start a Development or Sandbox sessions, the "Emulation-Passed" or "Emulation-Failed" flag is cleared.

Testbed Execution: A user can submit a "Saved" experiment for testbed execution only if the user currently has an "Emulation-Passed" flag. The saved experiment is scheduled for testbed execution at a time convenient and feasible for the platform; the Testbed mode is again both scheduled and batch. During testbed execution, the Experiment Oversight system monitors the site the experiment is executing on, and may abort the experiment if safety violations appear to have occurred or be imminent. The experiment reverts to Saved mode when (i) it completes normally, (ii) it is aborted, or (iii) the scheduled duration for the experiment runs out.

Users are expected to develop code for the purpose of collecting experimental data, and save it "locally" on their E-VMs. The saved E-VM image is the only way for users to receive data out of their experimentation; the platform does not auto-generate any data for the user. Upon successful completion of the experiment, the user receives a notification to trigger another Development session at their convenience. When they do, the E-VMs created in the virtual environment contain any output saved by the experimenter's code, which the experimenter can thus review.

If a testbed execution is aborted, enough information will be provided to the user by the system so that the user can meaningfully re-develop their experiment to avoid the abort at a later attempt.

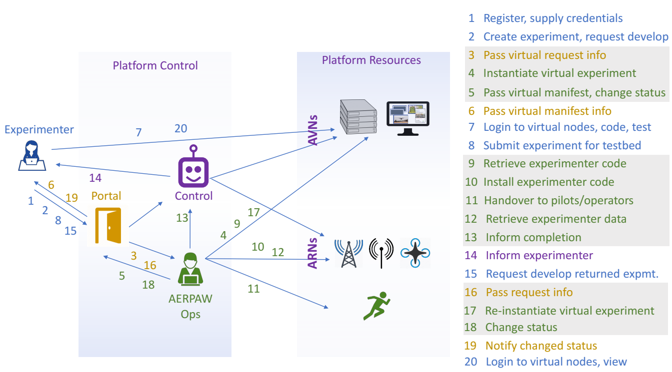

Fig. 4: AERPAW Canonical Workflow Details

Fig. 4 presents a diagrammatic self-explanatory view of the various steps in a typical Create-Develop-Submit-Review cycle for a canonical experiment. More details about the individual steps are available at the AERPAW User Manual [16].

B. Non-Canonical Workflows

The canonical mode of operation cannot automatically encompass the 3PBBE we mentioned in Section IV, because of the lack of (i) virtual computational models of such 3PBBE that can be incorporated into AERPAW's Development or Emulation environments, (ii) programmable control/configuration capabilities, to enable batch operation of Experimenter code (without live console access). Designing and developing solutions to these are outside the scope or resources of the current AERPAW project. Accordingly, we do not expect to provide 3PBBE access through the canonical workflow. Nevertheless, we understand that AERPAW can bring significant value to experimenters by providing access to such real-world equipment. The following non-canonical workflows can be used to provide interested and qualified experimenters with such access.

Experiment-as-a-Service (EaaS). This workflow is essentially identical to the canonical workflow, and can be used even for canonical experiments by experimenters who are not comfortable with SDR or UAV programming, and uninterested in learning. In this workflow, the experimenter simply describes (in documents, and through interaction with the AERPAW team) their experimental intent, and the experiment design, implementation, and execution, are all carried out by AERPAW personnel.

Although it is a much more protracted process than Program-it-Yourself, it has the advantage that non-programmable 3PBBE resources can be accommodated in the experiment without additional effort.

Experiment-Development-as-a-Service (EDaaS). This is an amalgamation of the canonical workflow and EaaS. As in EaaS, AERPAW personnel designs and develops the experiment based on experimenter intent, but once the base experiment is programmed, it is turned over to the experimenter who can further develop and customize it, to create future experiments. Beyond the handover, the experimenter follows the canonical workflow. However, if 3PBBE are included, then the experiment cannot return to the canonical workflow. For such experiments, the EDaaS workflow degenerates to the EaaS workflow.

Live Limited Access. By going through some technical material and cooperative agreements, after some initial level-setting, qualified experimenters can be provided with testbed sessions in which they will have live login access to a specific predetermined subset of 3PBBE. Such use will not include the ability to integrate canonical experiment resources (i.e. SDR nodes and UAVs/UGVs) into such live access.

Mixed and Custom Service Approaches. If none of the above approaches can accommodate some particular, complex requirements of an experimenter, the AERPAW team will be open to discussing and coming up with unique custom approaches to support such requirements.

Bring Your Own Device (BYOD). A particular workflow of interest to experimenters developing or studying custom or cutting-edge hardware pertains to BYOD possibilities. Note that although AERPAW is receptive to the concept of BYOD, our goal is to be able to manage the process so that the platform's ability to verify and enforce safe operations, staying within appropriate regulations, is not compromised. Our BYOD process requires the experimenter proposing a BYOD scenario to provide specific information regarding the equipment (physical, electrical, software, etc.) that enable us to assess the feasibility of doing so. Regarding workflow, we note that BYOD cases are natural fits for one or the other non-canonical workflows. For example, some researchers interested in passive signature-based drone detection may want to specify that AERPAW personnel should fly a certain model of drone (or send us their own custom drones) in our wireless field, to test detection algorithms. Such a drone would typically not be programmable, but we would still be able to manually execute the experiment. We do not envisage being able to, or commit to, providing a canonical workflow for BYOD equipment.

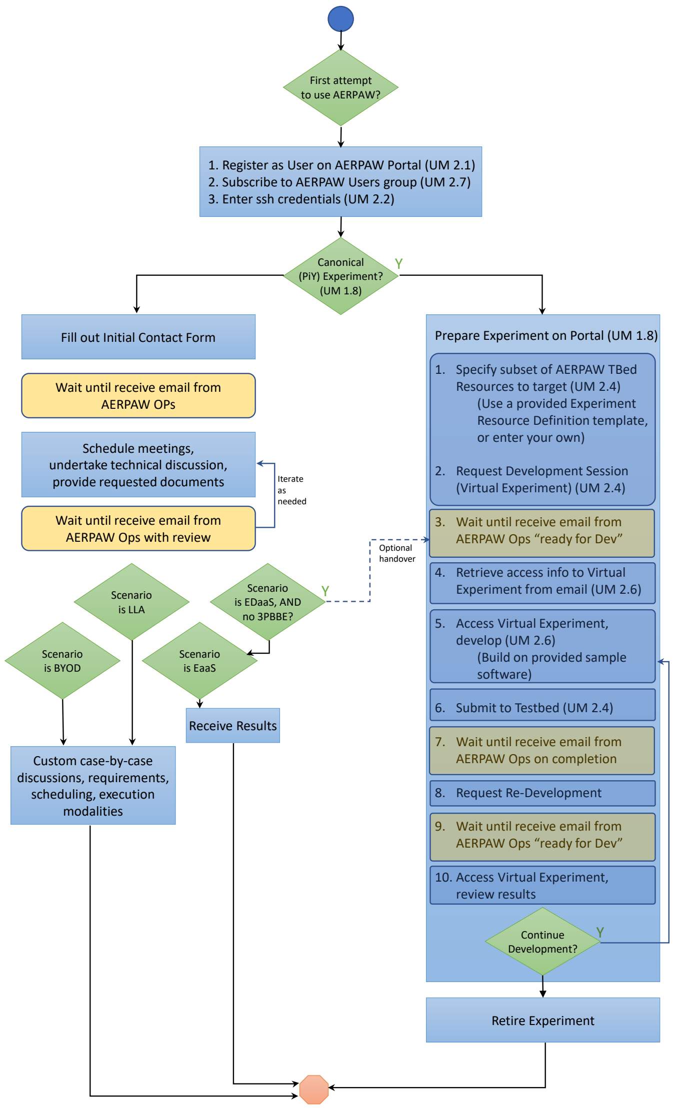

Fig. 5: Planning an AERPAW Experiment

Fig. 5 shows a summary view of starting AERPAW use. The numbers refer to sections of the AERPAW User Manual [16] that contain more detailed information about each step.

Acknowledgment

This material presented in this paper is based upon work supported by the National Science Foundation PAWR Program under Grant No.: CNS-1939334. We also gratefully acknowledge colleagues on the AERPAW team, and other colleagues in the community, who helped articulate and crystallize research needs, and meaningful usage models, for AERPAW's original and ongoing testbed vision. We gratefully acknowldge the industry partners who provided equipment as in-kind support to be used in AERPAW as 3PBBE, and allowed their names to be used on the AERPAW website, in this paper, and elsewhere.

VI. Conclusion

We have presented an overview of AERPAW usage, with context that provides insight into the motivation, goals, and constrains for the Testbed workflow design. To understand the Testbed usage model, we have also provided the architecture of the different entities that their interactions in the Testbed to support both canonical and non-canonical workflows. We hope that this will enable a large variety of researchers to utilize AERPAW to advance their valuable research.

References

[1] SESAR, European Union, Euro Control, "European drones outlook study: Unlocking the value for Europe," Nov. 2016. [Online]. Available: {https://www.sesarju.eu/sites/default/files/documents/reports/European_Drones_Outlook_Study_2016.pdf}
[2] AUVSI, "The Economic Impact of Unmanned Aircraft Systems Integration in the United States," Mar. 2013. [Online]. Available: https://higherlogicdownload.s3.amazonaws.com/AUVSI/958c920a-7f9b-4ad2-9807-f9a4e95d1ef1/UploadedImages/New_Economic%20Report%202013%20Full.pdf
[3] Goldman Sachs, "Drones: Reporting for Work." [Online]. Available: https://www.goldmansachs.com/insights/technology-driving-innovation/drones/
[4] J. Desjardins, "Amazon and UPS are betting big on drone delivery," Business Insider, Mar. 2018. [Online]. Available: https://www.businessinsider.com/amazon-and-ups-are-betting-big-on-drone-delivery-2018-3
[5] "Amazon Prime Air Drone Delivery." [Online]. Available: https://www.amazon.com/Amazon-Prime-Air/b?ie=UTF8&node=8037720011
[6] "Google Project Wing." [Online]. Available: https://x.company/projects/wing/
[7] "Uber Elevate." [Online]. Available: https://www.uber.com/us/en/elevate/
[8] J. Johnsson and A. Levin, "Boeing is getting ready to sell flying taxis," Bloomberg, Mar. 2018. [Online]. Available: https://www.bloomberg.com/news/articles/2018-03-01/boeing-is-getting-ready-to-sell-flying-taxis-within-a-decade
[9] "Zipline medical supply drone delivery." [Online]. Available: http://www.flyzipline.com/
[10] "Flytrex on demand drone delivery." [Online]. Available: http://www.flytrex.com/
[11] "Matternet delivery drones." [Online]. Available: https://mttr.net/
[12] Art Pregler, AT&T, "When COWs Fly: AT&T Sending LTE Signals from Drones," Feb. 2017. [Online]. Available: https://about.att.com/innovationblog/cows_fly
[13] Vodafone, "Beyond visual line of sight drone trial report," Nov. 2018. [Online]. Available: https://www.vodafone.com/content/dam/vodafone-images/media/Downloads/Vodafone_BVLOS_drone_trial_report.pdf
[14] X. Lin, V. Yajnanarayana, S. D. Muruganathan, S. Gao, H. Asplund, H.-L. Maattanen, M. Bergstrom, S. Euler, and Y.-P. E. Wang, "The sky is not the limit: LTE for unmanned aerial vehicles," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 204–210, 2018.
[15] Qualcomm, "LTE Unmanned Aircraft Systems," May 2017. [Online]. Available: https://www.qualcomm.com/media/documents/files/lte-unmanned-aircraft-systems-trial-report.pdf
[16] "AERPAW User Manual." [Online]. Available: https://sites.google.com/ncsu.edu/aerpaw-wiki/
[17] "Aerial Experimentation and Research Platform for Advanced Wireless." [Online]. Available: https://aerpaw.org
[18] A. Panicker, O. Ozdemir, M. L. Sichitiu, I. Guvenc, R. Dutta, V. Marojevic, and B. Floyd, "Aerpaw emulation overview and preliminary performance evaluation," *Computer Networks*, vol. 194, p. 108083, 2021.
[19] V. Marojevic, I. Guvenc, R. Dutta, M. L. Sichitiu, and B. A. Floyd, "Advanced wireless for unmanned aerial systems: 5g standardization, research challenges, and aerpaw architecture," *IEEE Vehicular Technology Magazine*, vol. 15, no. 2, pp. 22–30, 2020.