

Introducing Thuraya IP

While Internet access is already a commodity in Europe, there are still many places in the world without proper internet access. For these "off-the-grid" locations, [Thuraya IP](#) can provide internet access (upstream/downstream of 384kb/444kb) using their durable satellite broadband terminals. With the common use case for Thuraya IP surely being much more "exotic" than what we normally handle at Nexedi, we still happened to be working on a project which required ERP5 to run on remote locations in China with no reliable internet access whatsoever - so we gave Thuraya a try.



The Challenge - Keep Traffic to a Minimum

Using a service such as Thuraya comes at a hefty price tag per Megabyte, so we had to design our application both ergonomic and economic and quickly opted for a [RenderJS](#) based HTML5 offline-capable ERP5 interface which stores and synchronizes data using [jIO](#), our virtual document database. The whole application was to be run on our experimental [NayuOS](#) ensuring as little data as possible is being sent through background processes and downloads.

RenderJS & jIO - A Match Made In The Cloud

With our web component framework [RenderJS](#) we were quickly able to generate the necessary UI from generic, reusable components and stored these in [application cache](#) keeping overall architecture at a minimum while making sure the app would be downloaded to a client on initial access only. On application level, [jIO](#) was setup as a replicate storage utilizing both [IndexedDB](#) and synchronizing to the central ERP5 instance. This way remote sites could work autonomously with data being replicated to ERP5 in regular intervals (currently triggered manually).



NayuOS - Private OS

Keeping users data private has always a top priority for Nexedi - even before operating systems like Windows 10 became [publicly scrutinized](#) for sending data to backend servers regardless of privacy settings. What started out as internal research project, resulting in our [first fully free Laptop](#) has turned into NayuOS, an operating system based on Android but free of most of the software connecting to Google services in the background. We envision NayuOS becoming our future operating system of choice and thus have a dedicated team working on NayuOS around the clock. For this project NayuOS was an easy choice, because we could actively control the data our application was exchanging while making sure as little data as possible was exchanged through OS background processes.



A Dry Run or "Time To Play"

Once we had a Thuraya device in hand it was time to do some real-world testing. We setup camp on different spots around the office and after coming to grips with the software, setting up our satellite broadband connection was as easy as pointing the device in the correct direction with the suggested elevation angle. We ran a couple of tests showing our applications initial load to be around 500kb. We wrote several sample records to the system and then synchronized with the main ERP5 server at less than 10kb per record. Other background traffic was neglectable and working with the ERP5 over satellite was a breeze. This meant at a pricetag of 4€ per Megabyte and an average of 100 records/day being synchronized our application could run over Thuraya at just above 100€ per month - a job well done and ERP5 optimized for use over Thuraya IP.



Beyond The Project

We were happy to see all of our software components meshing well together and having autonomous working units operating solely in the browser. Of course still limited in functional scope, this is the direction many of our research efforts are targeted to these days so stay tuned to our blog to see what else Nexedi is coming up with in the months ahead.