Files in this folder work with the package server and download the necessary versions to the local file-system.

Names

- Atmosphere - the packaging server UI, available at https://atmospherejs.com/.

- Troposphere - the actual package server that manages, stores and serves the binaries built for different architectures. It exposes an API that can be used by other front-ends (meteor tool, fastosphere, etc).

  lore: "Troposphere is the name of atmosphere's inner-most layer!"

- Warehouse - the name of the older downloaded packages layout that was used in pre-0.9.0 versions of Meteor before the official package server. Warehouse wasn't suitable for the new needs, so it was deprecated, but its ~~soul is still around~~ code is still used for backwards-compatibility.

- Tropohouse - the cross of "Troposphere" and "Warehouse" - a new file-system layout that is used in the post-0.9.0 world. Tropohouse stores compiled Isopacks on disk by package, by version, by architecture, by build.

- "The Red Pill" - a fake pre-0.9.0 format release that upgrades the Warehouse to Tropohouse: repo.

  lore: "You take the blue pill—the story ends, you wake up in your bed and believe whatever you want to believe. You take the red pill—you stay in Wonderland, and I show you how deep the rabbit hole goes." - Morpheus to Neo, Matrix.

The historical background

In the pre-0.9.0 world, the world was simple. The grass was greener, the trees were taller. We were younger. 3rd party packages were none of our concern. All 3rd party packages used to be distributed by a community-built tool called "Meteorite" or `mrt`. The packages catalog was on Atmosphere.

Every package used to be versioned with the release. Individual packages didn't have a version at all. Springboarding worked by loading a different release of Meteor entirely from a Warehouse folder.

In the newer world that happened with the release 0.9.0 on Aug 26 2014, MDG revealed the new package server that was optimized for serving tarballs, had versioning per package and was keeping track of builds for different architectures for packages with binary dependencies.

With that, the new set of commands in CLI came along for searching packages and viewing the individual info (`meteor search bla` and `meteor show <bla@1.2.3>`).

Internally, the on-disk storage switched to Tropohouse. Atmosphere started serving the packaging metadata based on Troposhere's API.

Meteor Tool changed the distribution from being the central piece every release is tied to, to a regular package with a special architecture `meteor-tool` (as opposed to `server` or `web.browser`).

Each package has a separate versioning line (although still optionally tied to releases).