

CHEATING GALL'S LAW



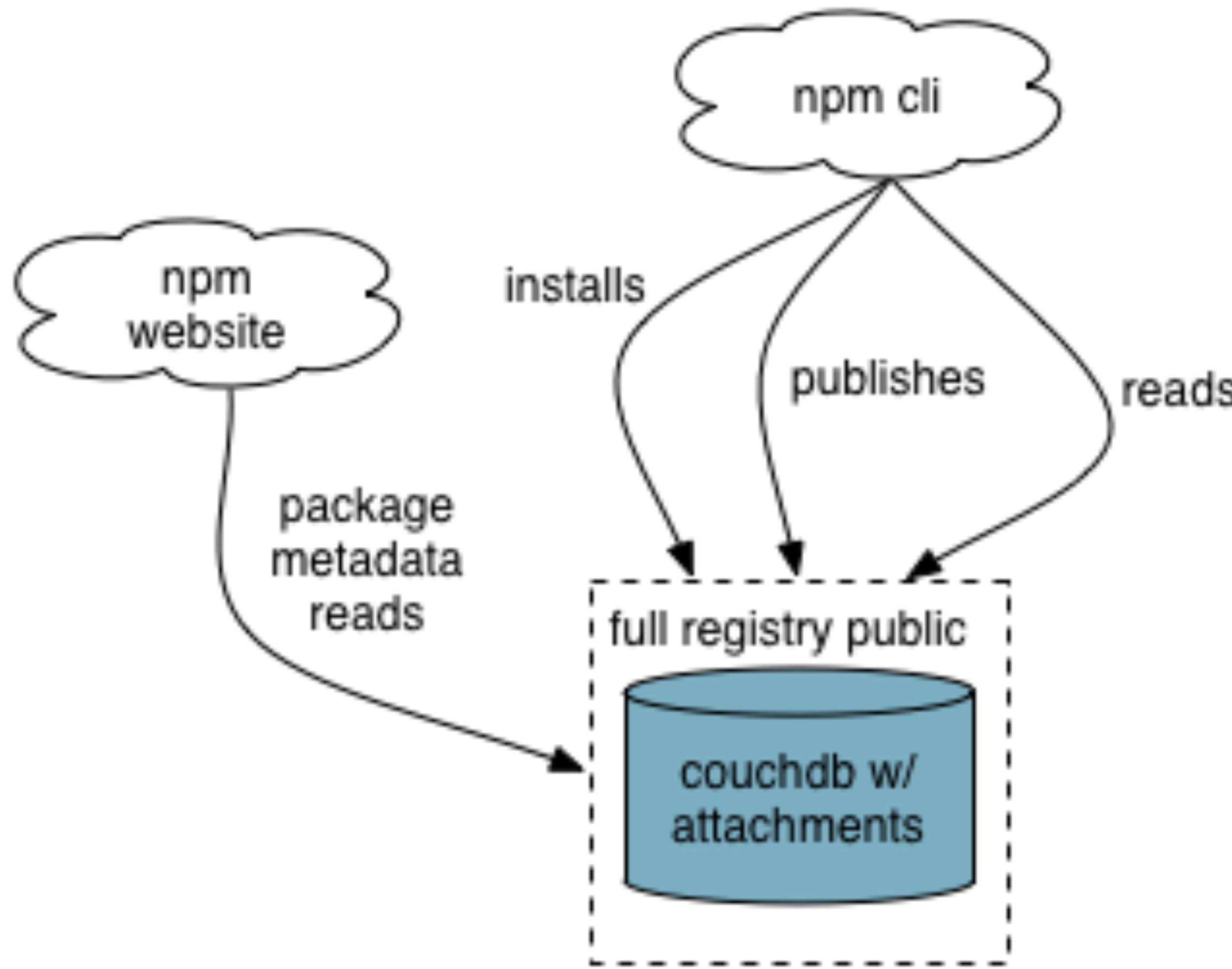
CJ SILVERIO
DIRECTOR OF ENGINEERING, NPM
@CEEJBOT

CHEATING GALL'S LAW HOW WE SPLIT A MONOLITH AND LIVED TO TELL THE TALE



MONOLITH

EVERYTHING IN ONE PROCESS



NPM'S MONOLITH: EMBEDDED IN COUCHDB

npm
website

installs

package

reads

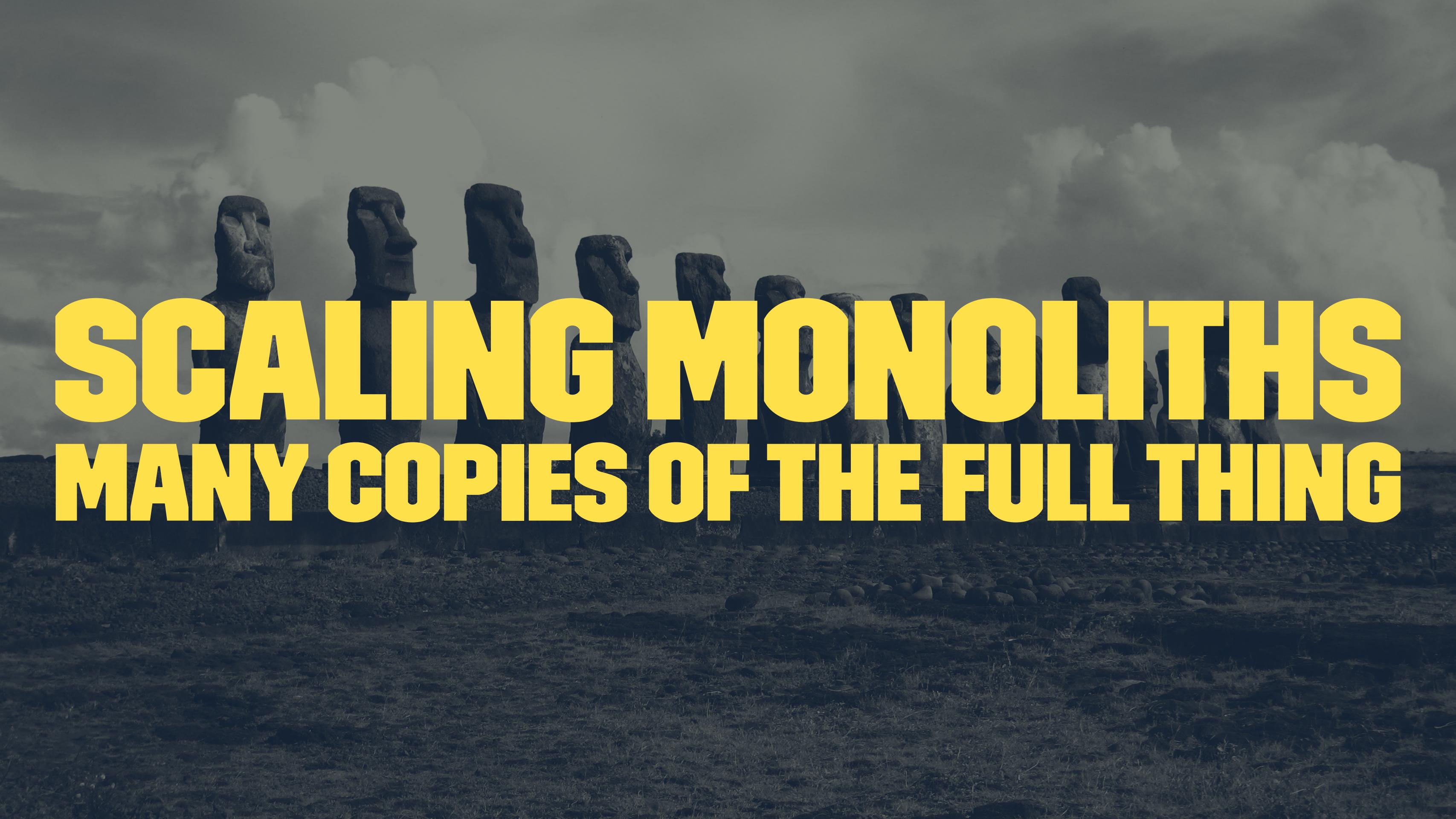
couchdb w/
attachments

**MONOLITHS
WORK JUST FINE**

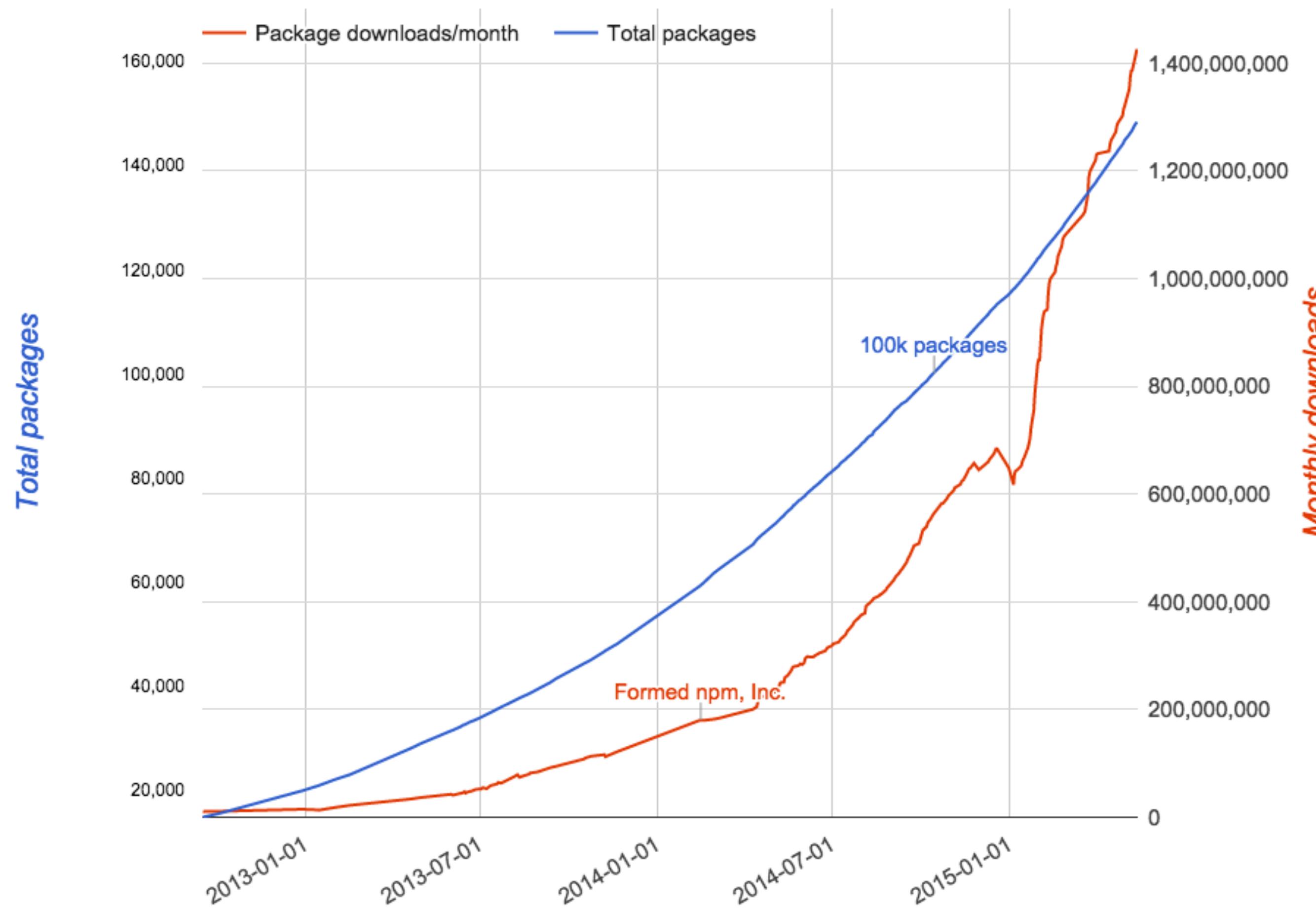
**WHATEVER IT TAKES
TO BUILD A SYSTEM
THAT SATISFIES YOUR USERS**

**SUCCESS!
NOW SCALE IT.**



A black and white photograph showing several large, monolithic stone statues (moai) standing in a row on a grassy hillside. The statues are carved from dark rock and have a simple, stylized human form with large heads and no visible arms or hands. They are set against a backdrop of a cloudy, overcast sky.

**SCALING MONOLITHS
MANY COPIES OF THE FULL THING**



A black and white photograph of the Moai statues on Easter Island. The statues are large, monolithic human figures with simple faces and no arms. They are silhouetted against a bright, cloudy sky. In the foreground, there's a dark, grassy field with some low-lying plants.

**EXPONENTIAL GROWTH OF NODE
RESULTED IN EXPONENTIAL GROWTH
OF THE NPM REGISTRY**

A black and white photograph showing several large stone Moai statues standing in a row on a grassy hillside. The statues are monolithic, carved from a single rock. They have a distinctive style with large, rounded heads and no visible bodies or arms. The background is filled with a dense, cloudy sky.

**EXPONENTIAL MONOLITHS
WERE GOING TO BE EXPENSIVE**

SPLITTING THE MONOLITH

YAY MICROSERVICES?

**JUST REWRITE EVERYTHING!
WHAT'S THE PROBLEM?**

**YOUR MONOLITH IS COMPLEX.
A SPLIT SYSTEM IS MORE COMPLEX.**

GALLO'S LAW
A COMPLEX SYSTEM THAT WORKS IS
INVARIABLY FOUND TO HAVE EVOLVED
FROM A SIMPLE SYSTEM THAT WORKED.

SYSTEMANTICS: HOW SYSTEMS REALLY WORK AND HOW THEY FAIL

GALL'S LAW

“A complex system that works is invariably found to have evolved from a simple system that worked. A complex system designed from scratch never works and cannot be patched up to make it work. You have to start over with a working simple system.”

**"A SIMPLE SYSTEM
MAY OR MAY NOT WORK!"**

GALLS LAW
SAYS WE CAN'T WRITE OUR
COMPLEX REPLACEMENT

**HOW DO YOU SPLIT
A MONOLITH
SUCCESSFULLY?**

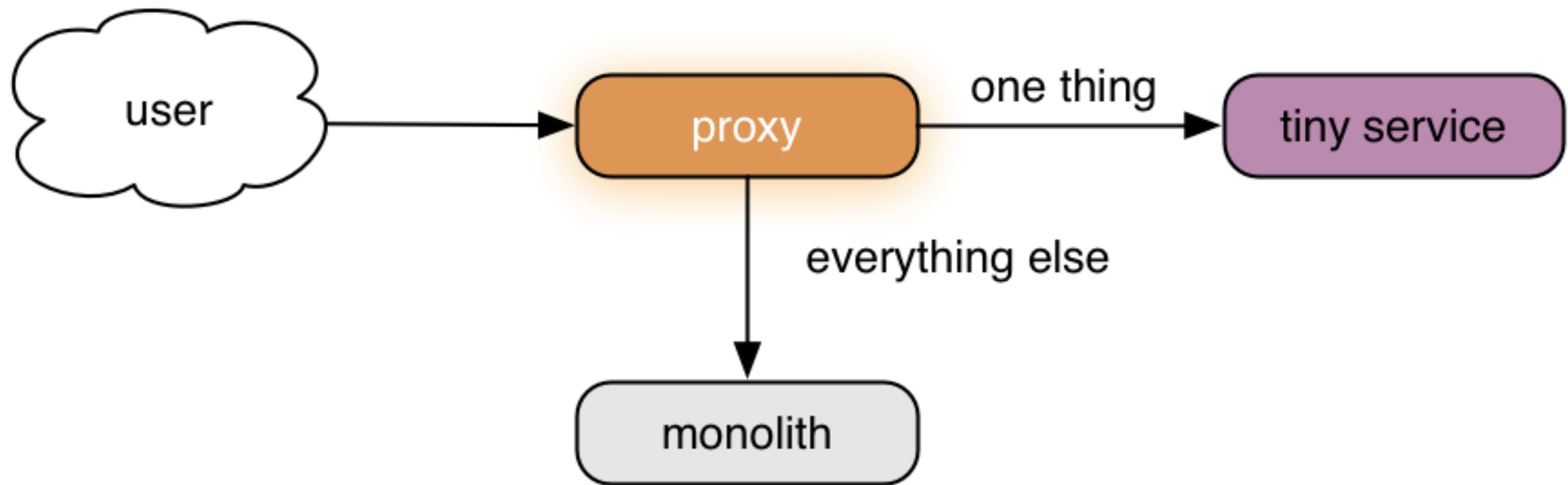
LET'S CHEAT.

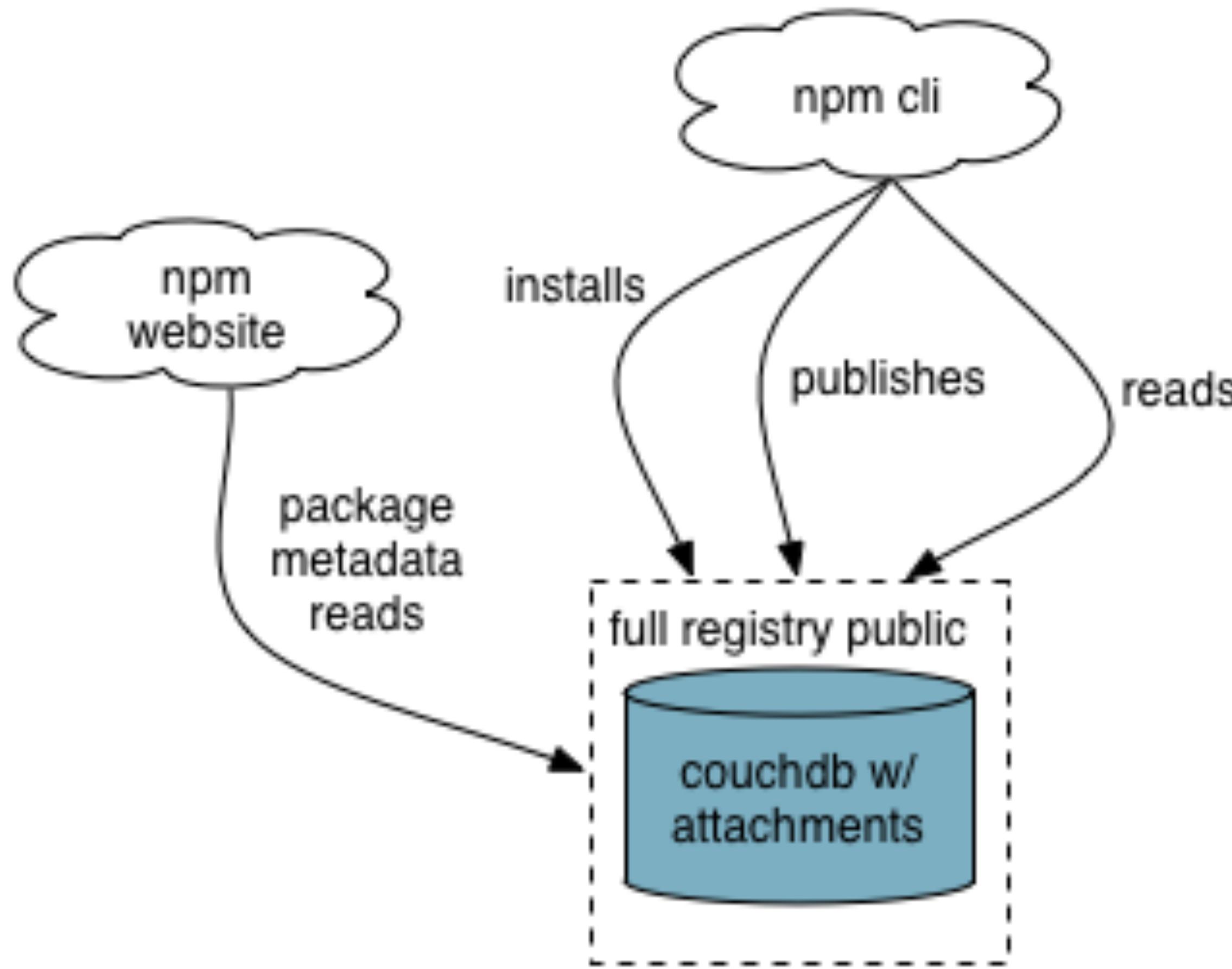
**Q: HOW DO YOU CHEAT?
A: BY NOT REWRITING THE WHOLE THING.**

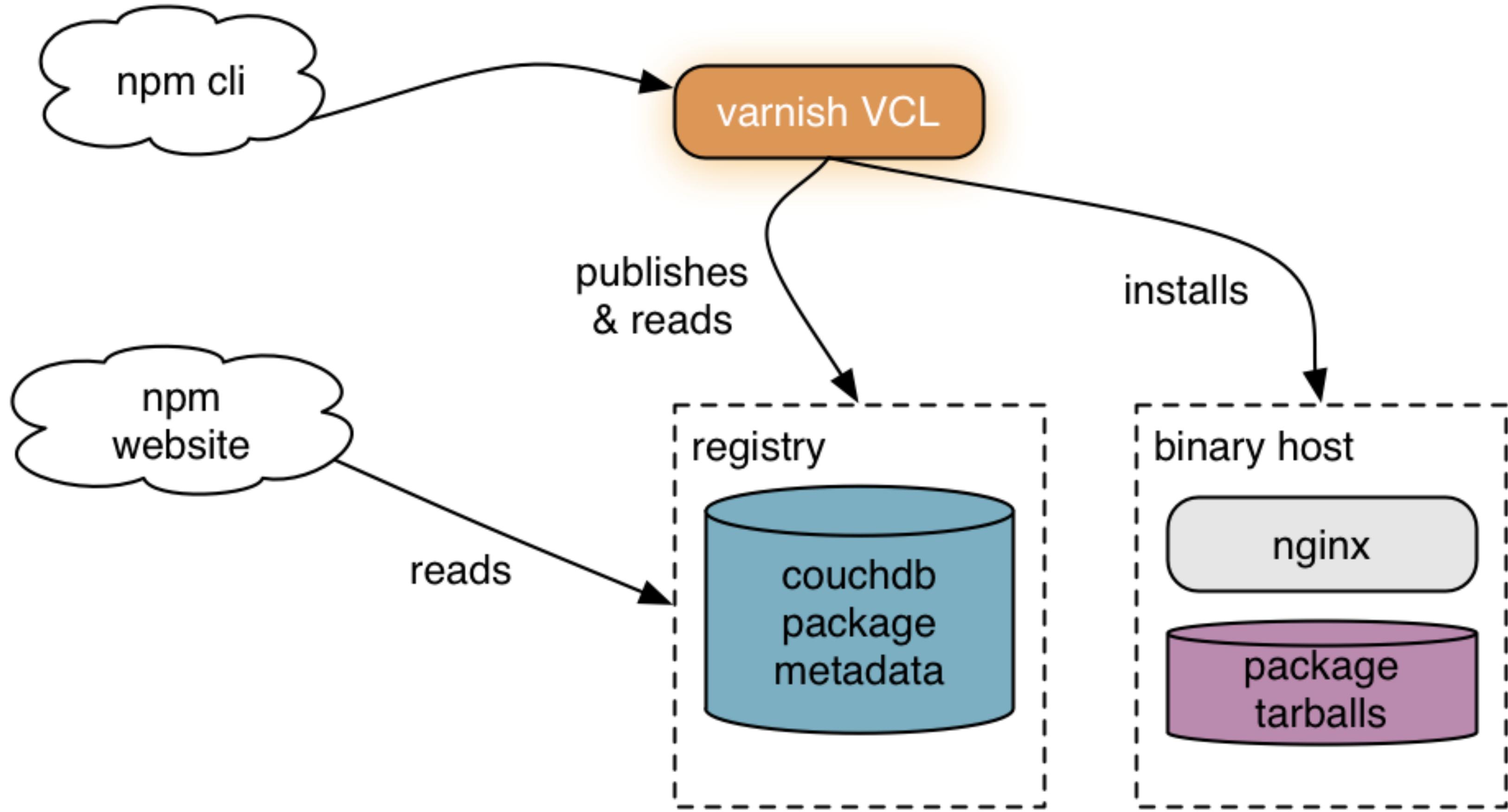
**SLICE OFF A PART OF THE SYSTEM
INTO A MODULE
WITH A CLEARLY-DEFINED INTERFACE**

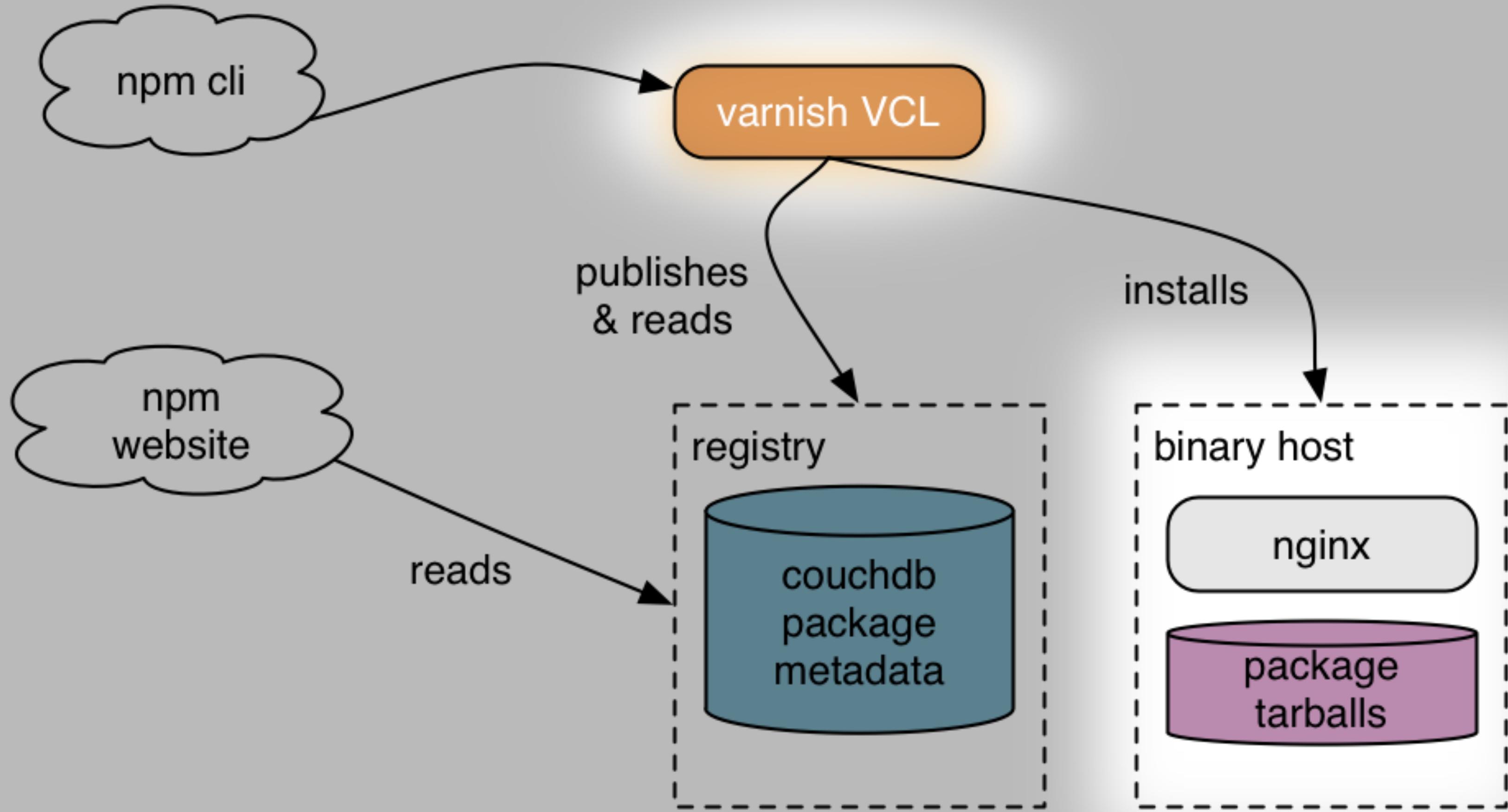
**THEN WRITE A
SECOND IMPLEMENTATION
OF THAT INTERFACE**

**SPLIT YOUR TRAFFIC
WITH A PROXY**

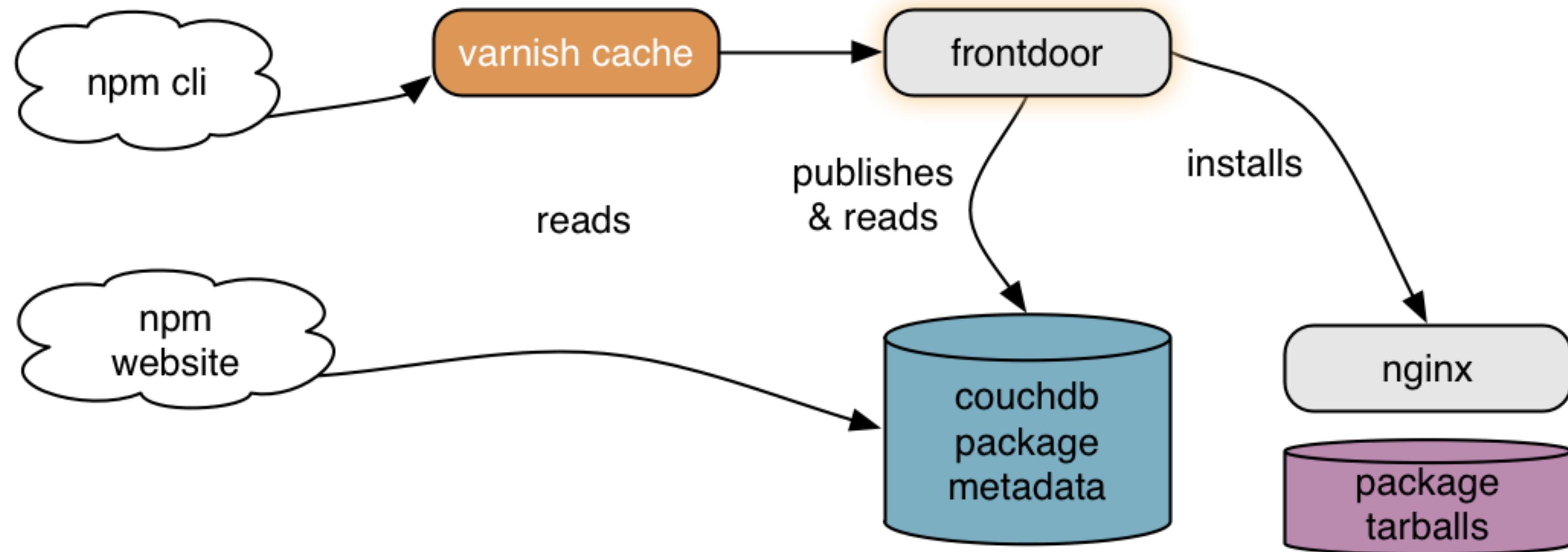


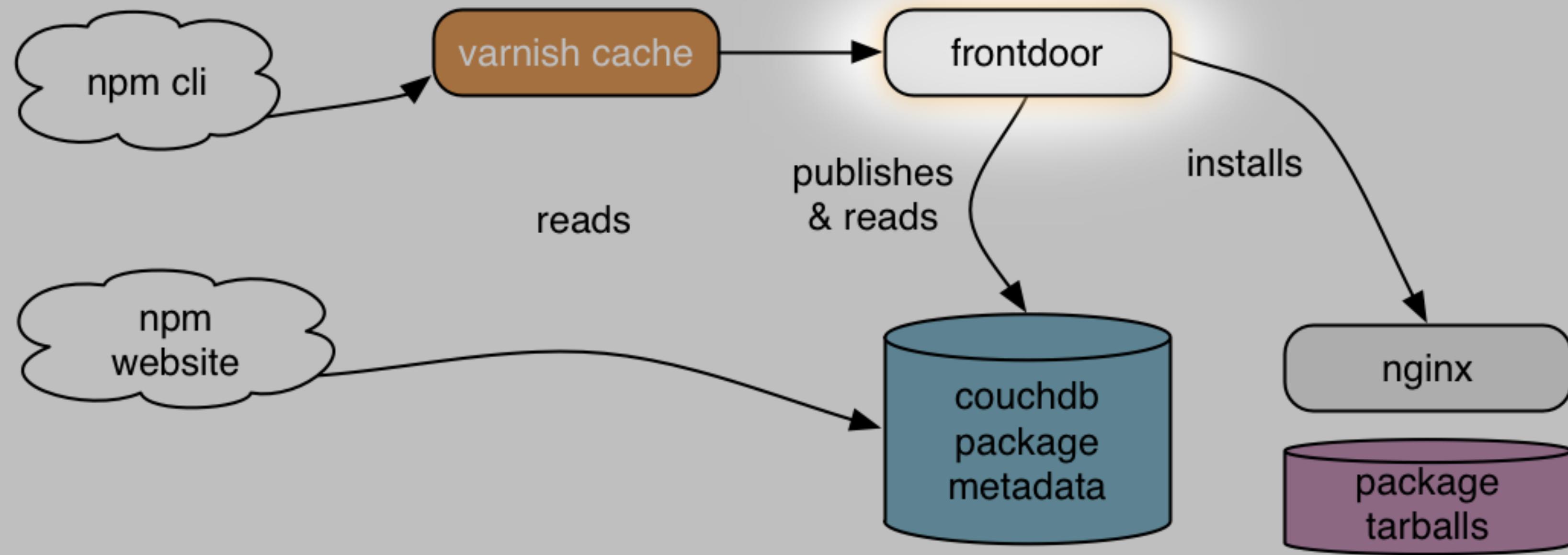






**NOW LET'S MAKE OUR PROXY SMART
REPLACE VARNISH
WITH A NODE SERVICE**

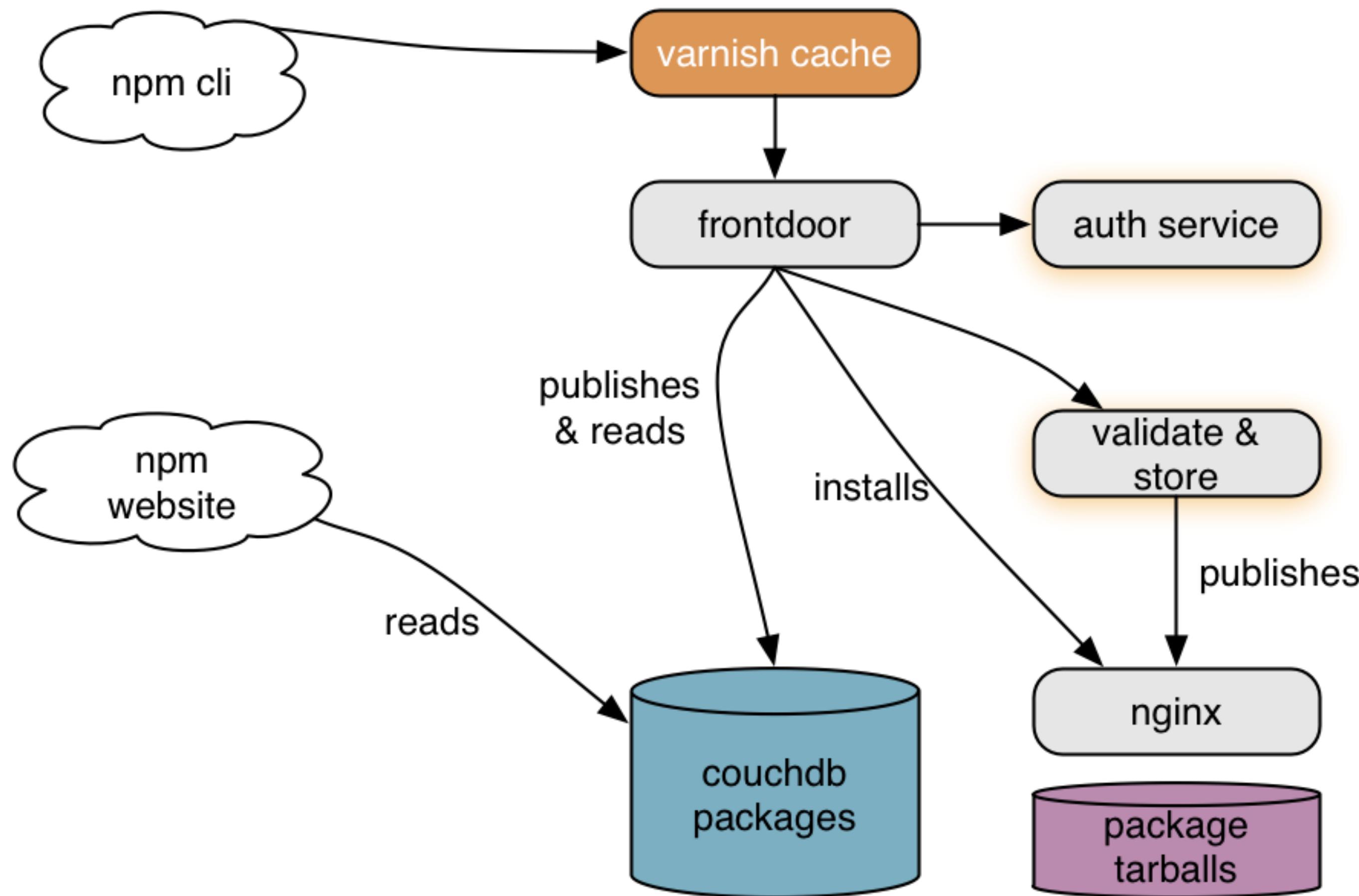


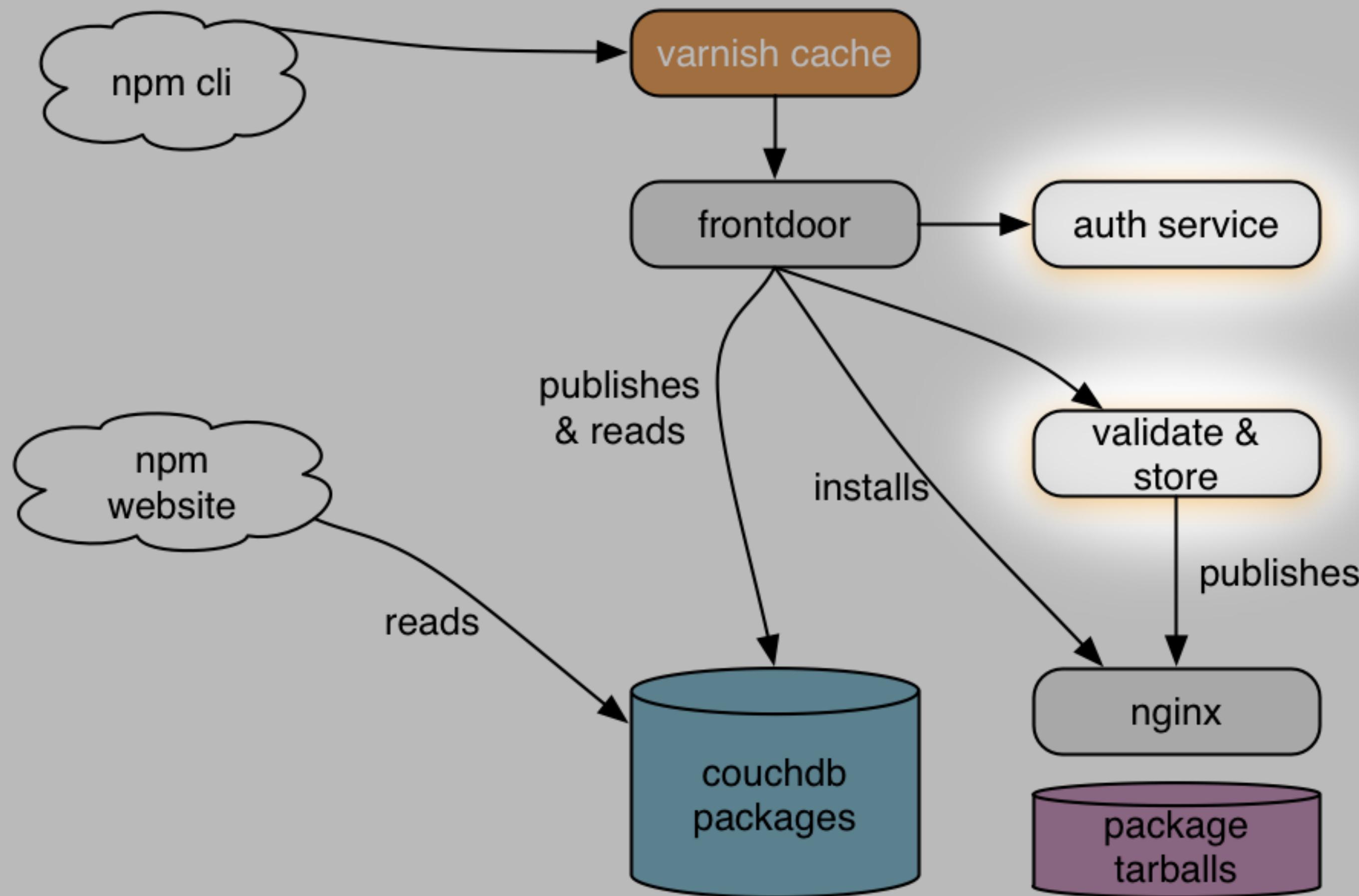


**FIRST STEP
JUST SEND EVERYTHING THROUGH**

**RECORD WHAT THE PROXY DOES
HOW DO PEOPLE USE YOUR SERVICE?
DISCOVER ITS TRUE BEHAVIOR**

**SECOND STEP
DIVIDE & CONQUER**



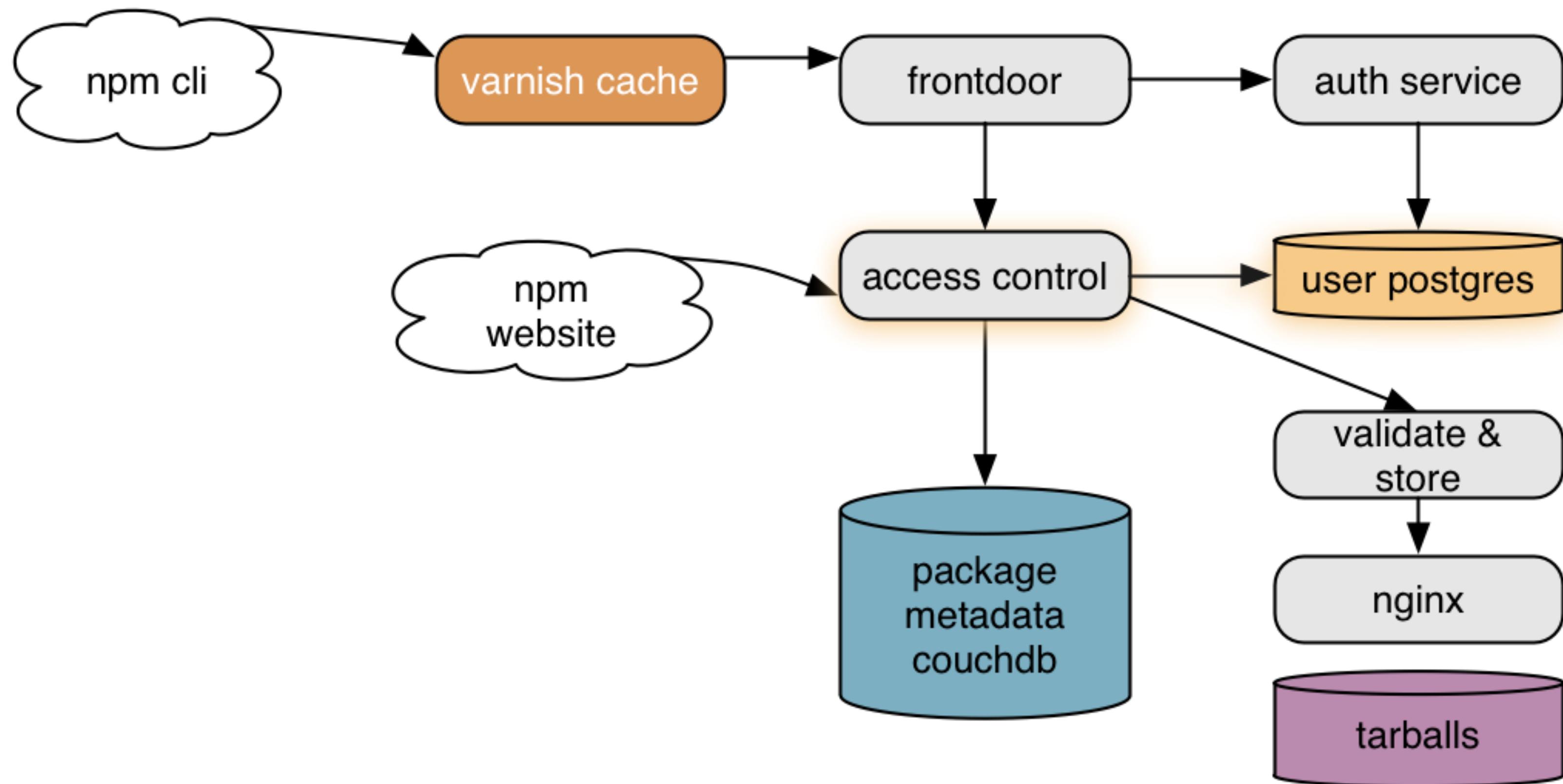


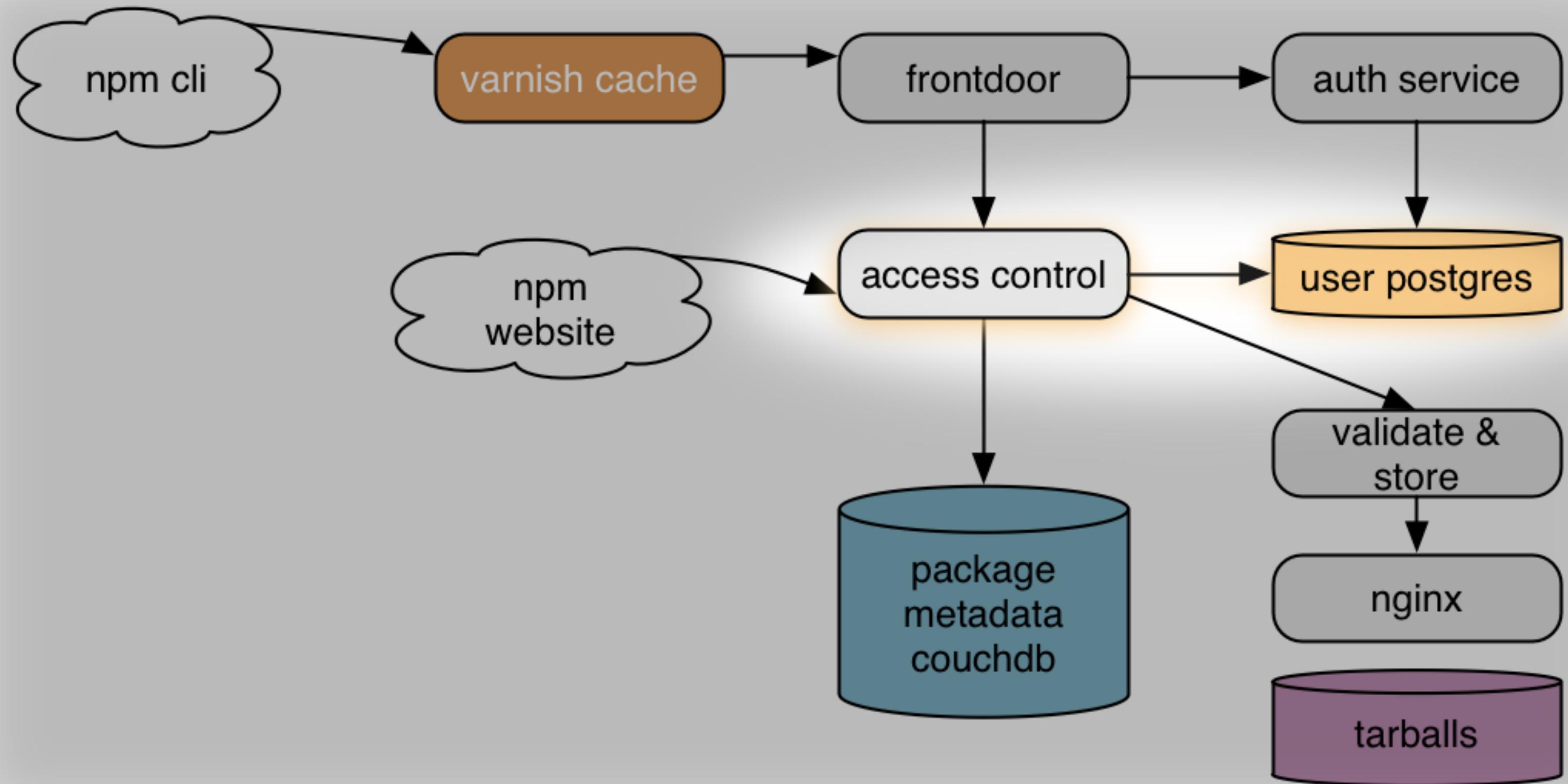
**BEAT GALL'S LAW WITH
MODULARITY
AKA INFORMATION HIDING**

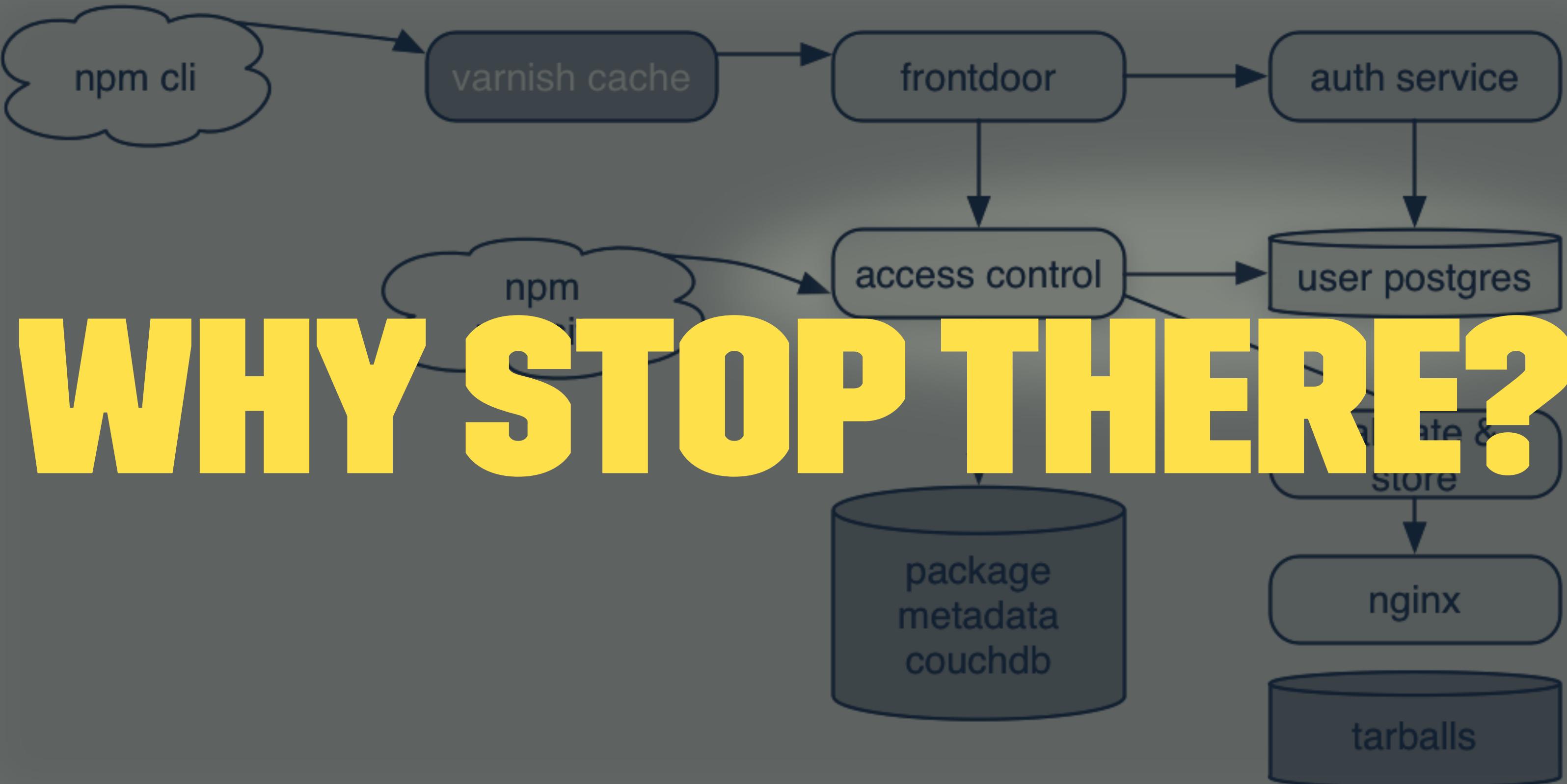
**EACH SERVICE IS A MODULE
A SIMPLE, TESTABLE SYSTEM
WHEN VIEWED IN ISOLATION**

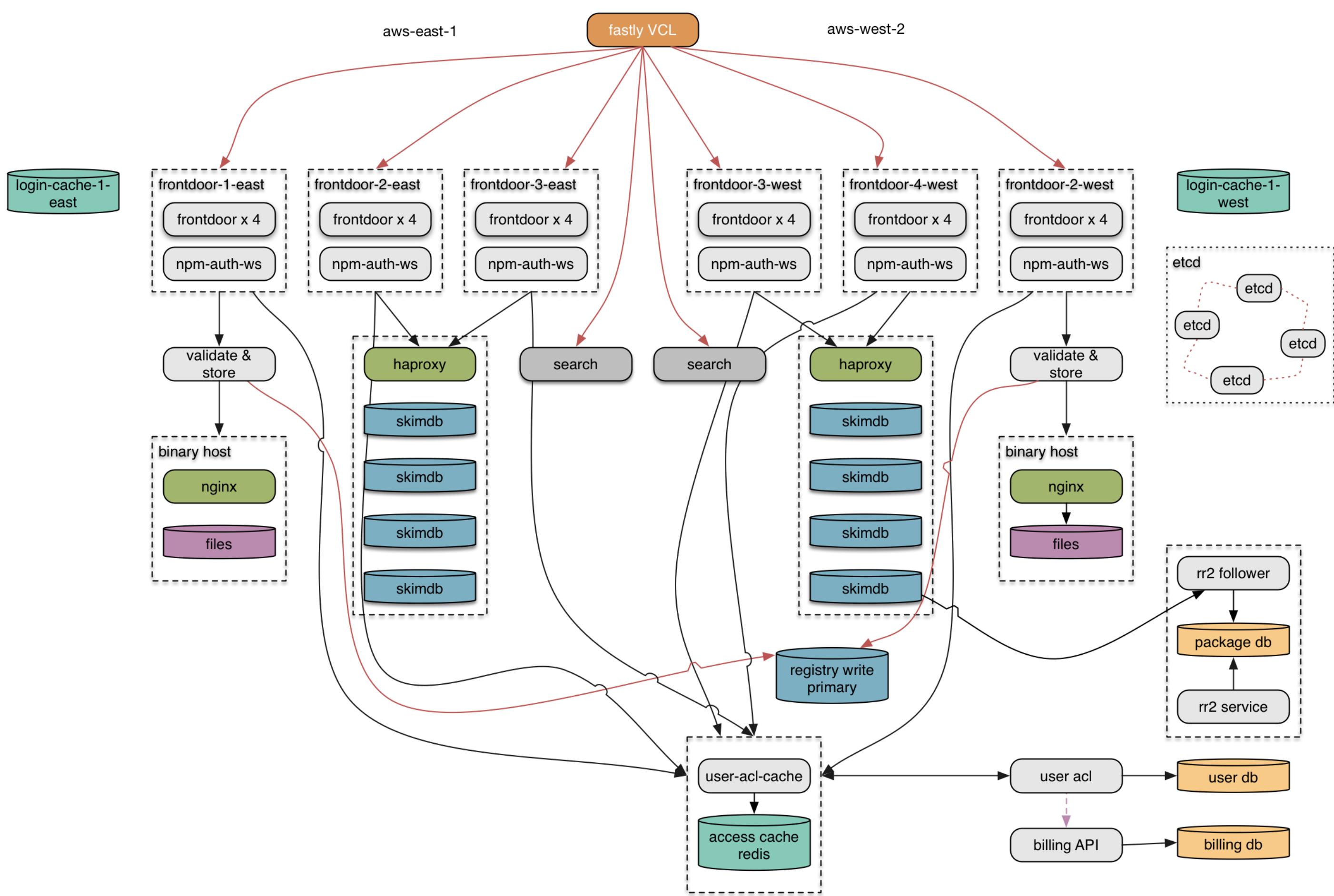
**BEST PART
NOW YOU CAN CHANGE
EVERYTHING**

**YOUR PLATFORM
YOUR DATABASE**









THE AWESOME

EACH PIECE IS SIMPLE

**MORE SCALING DIALS TO TURN
FINER-GRAINED PERF DATA**

**THE SYSTEM OVERALL IS COMPLEX
BUT LOOSELY COUPLED**

**YOU HAVE A WORKING SYSTEM
EVERY STEP OF THE WAY**

**A PROXY SERVICE LETS YOU
FLIP BACK & FORTH
BETWEEN IMPLEMENTATIONS**

THE PITFALLS

**NOW YOU'VE GOT A
DISTRIBUTED SYSTEM**

@ARCHITECTCLIPPY SAYS

“I see you have a poorly structured monolith. Would you like me to convert it into a poorly structured set of microservices?”

SECOND SYSTEM SYNDROME

"THIS TIME WE'LL DO IT RIGHT"

GENERALIZE ONLY WHEN FORCED

**PUT THIS TECHNIQUE
INTO YOUR TOOLBOX**

**SIMPLE WORKING SERVICE FIRST
SCALE IT LATER**

BE BOLD

YOU CAN CHANGE YOUR SYSTEM

**NOBODY NOTICED
WHEN WE CHANGED
THE ENTIRE NPM REGISTRY**

**PROXIES ARE COOL.
I'M INTO PROXIES NOW.**

NPM LOVES YOU