

**CHEATING GALL'S LAW**



**CJ SILVERIO**  
DIRECTOR OF ENGINEERING, **OPM**  
**@CEEJBOT**

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

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

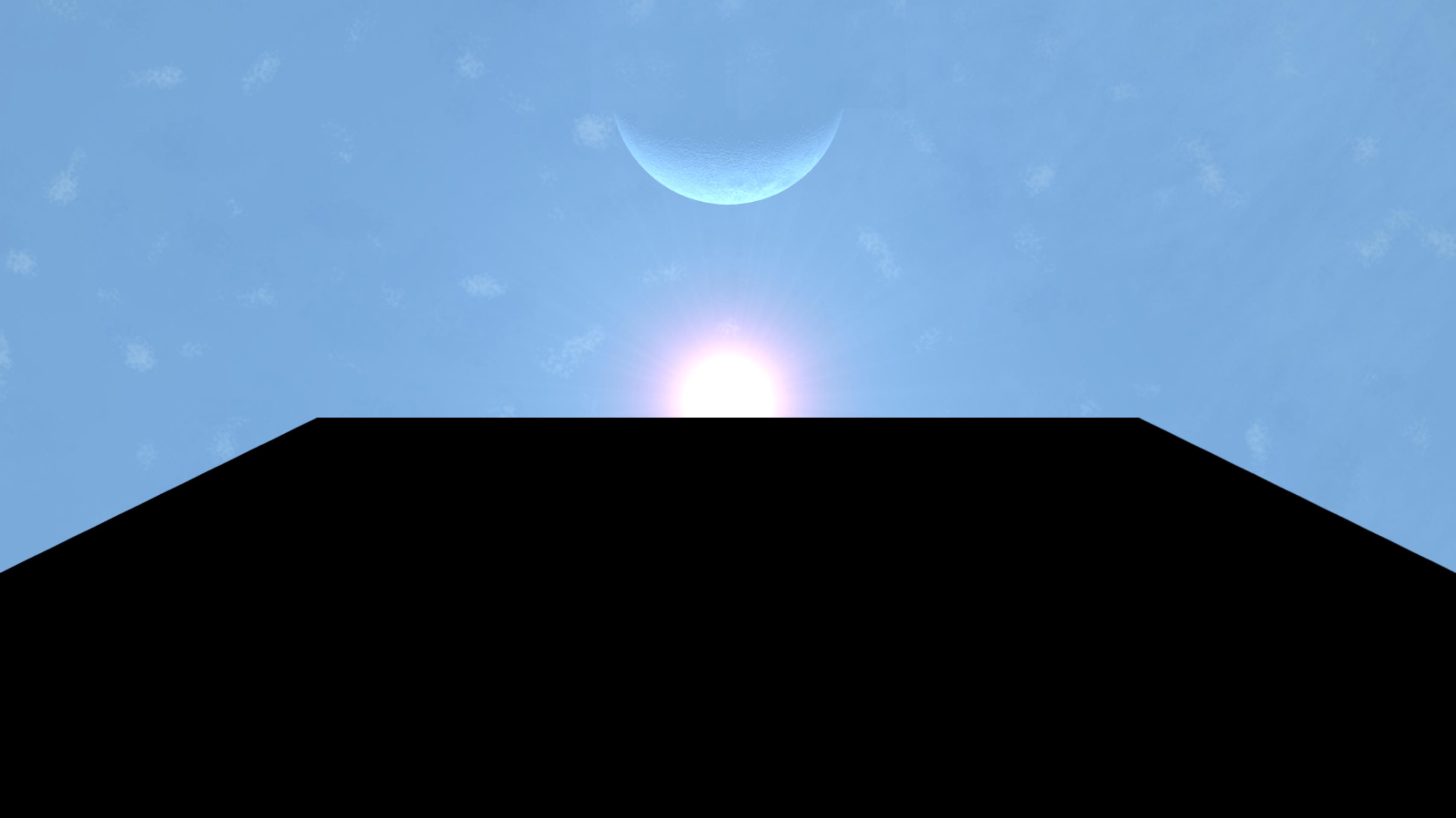
# GALL'S LAW IN FULL

“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!"**

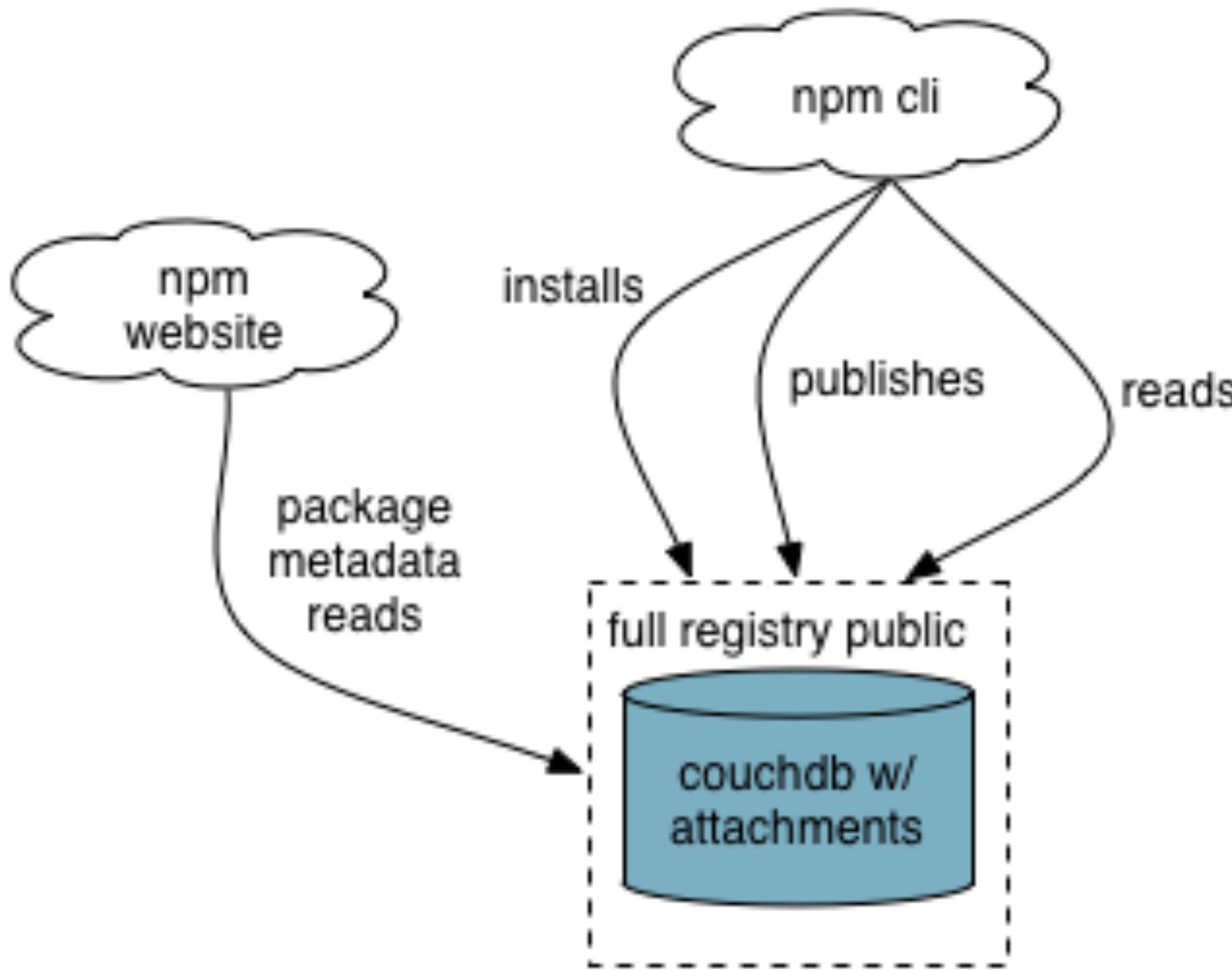
# **SYSTEMANTICS: HOW SYSTEMS REALLY WORK AND HOW THEY FAIL**

**STARTING SIMPLE  
OFTEN MEANS STARTING WITH A  
MONOLITH**



# MONOLITH

## EVERYTHING IN ONE PROCESS



# NPM'S MONOLITH: EMBEDDED IN COUCHDB

npm  
website

installs

package

reads

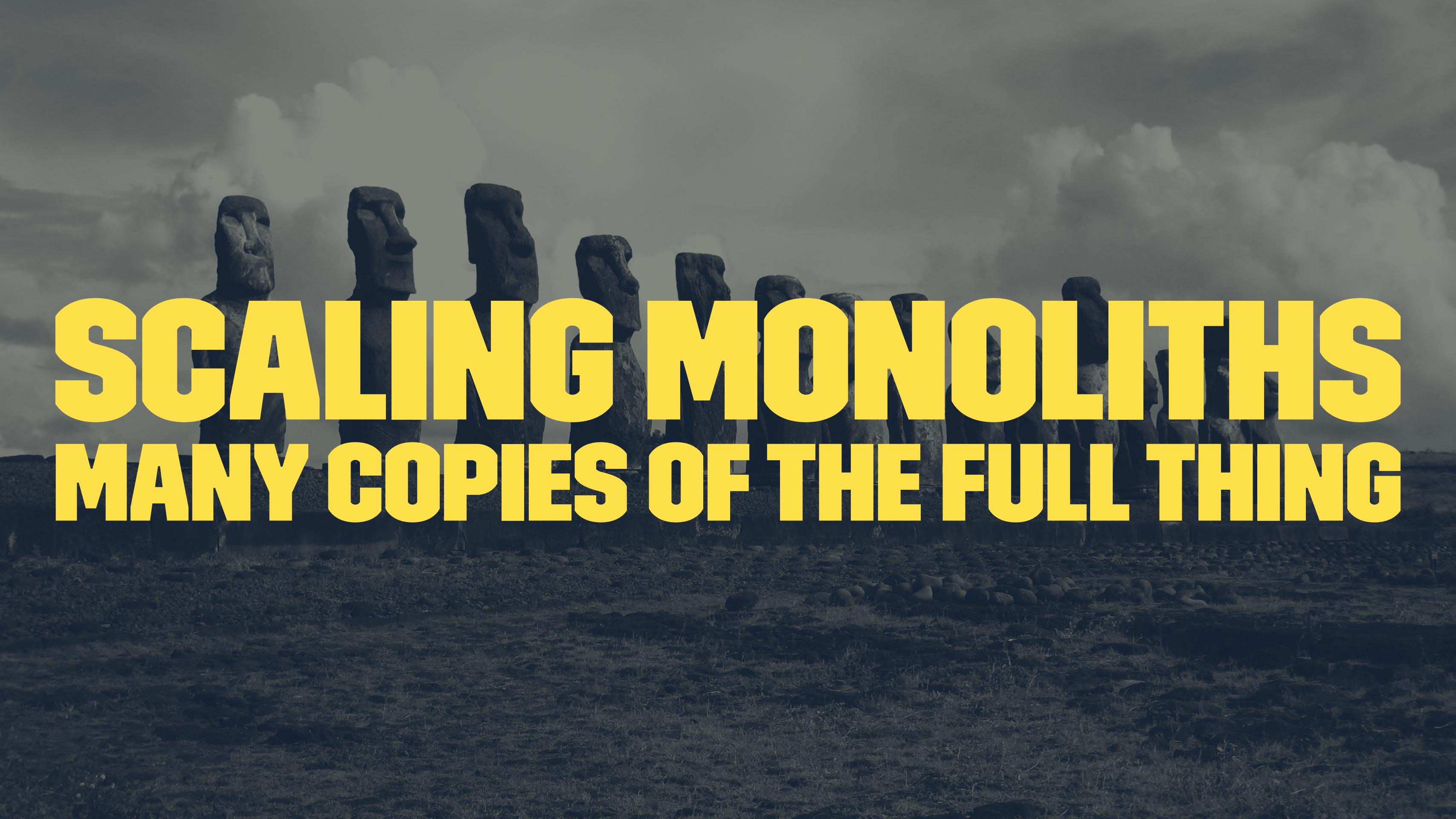
couchdb w/  
attachments

**MONOLITHS  
WORK JUST FINE**

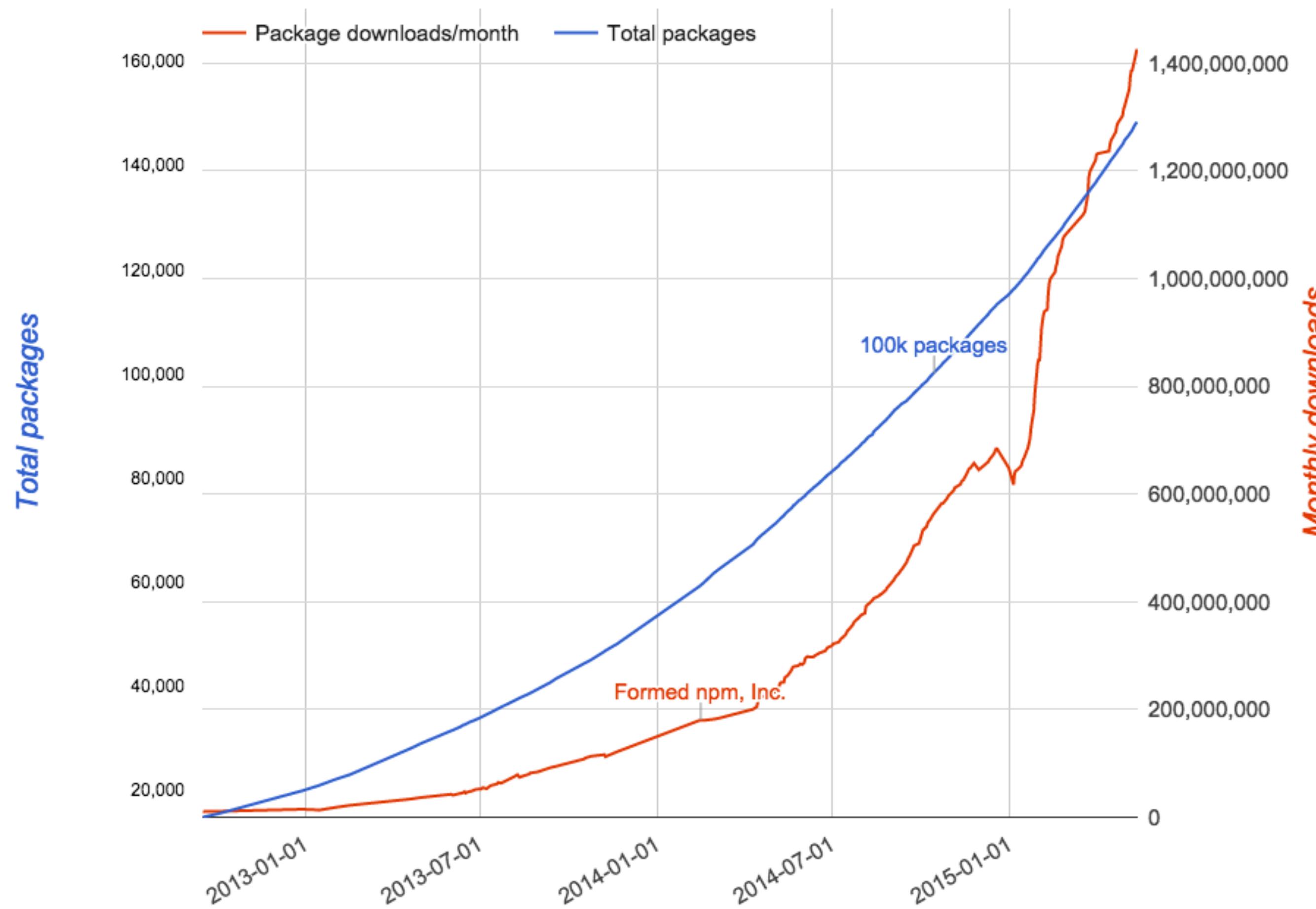
**WHATEVER IT TAKES  
TO BUILD A SYSTEM  
THAT SATISFIES YOUR USERS**

**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. 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 from the Easter Islands. They are standing upright in a row, facing slightly to the left. The background is filled with a dense, cloudy sky. The foreground is dark and appears to be covered in low-lying vegetation or rocks.

**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.**

**WHAT DID THAT GALL GUY SAY ABOUT  
COMPLEX WORKING SYSTEMS?**

**"IF A SYSTEM IS WORKING,  
LEAVE IT ALONE.  
DON'T CHANGE ANYTHING!"**

**GALVAN LAW  
WARNS US ABOUT  
THAT REWRITE**

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

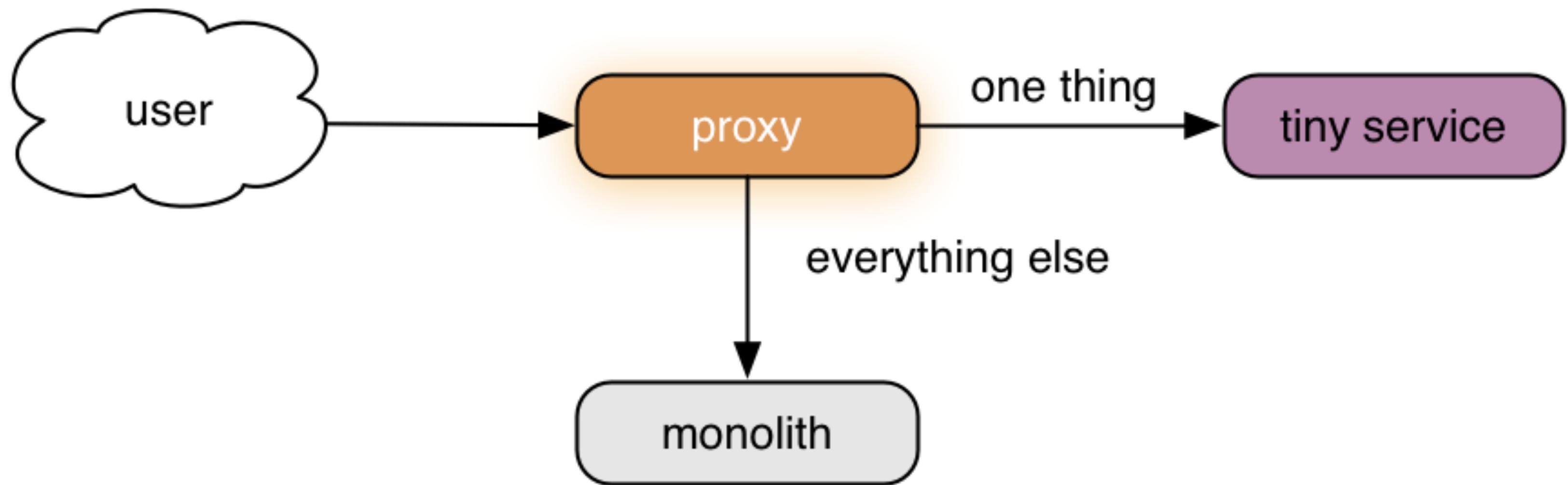
**LET'S CHEAT.**

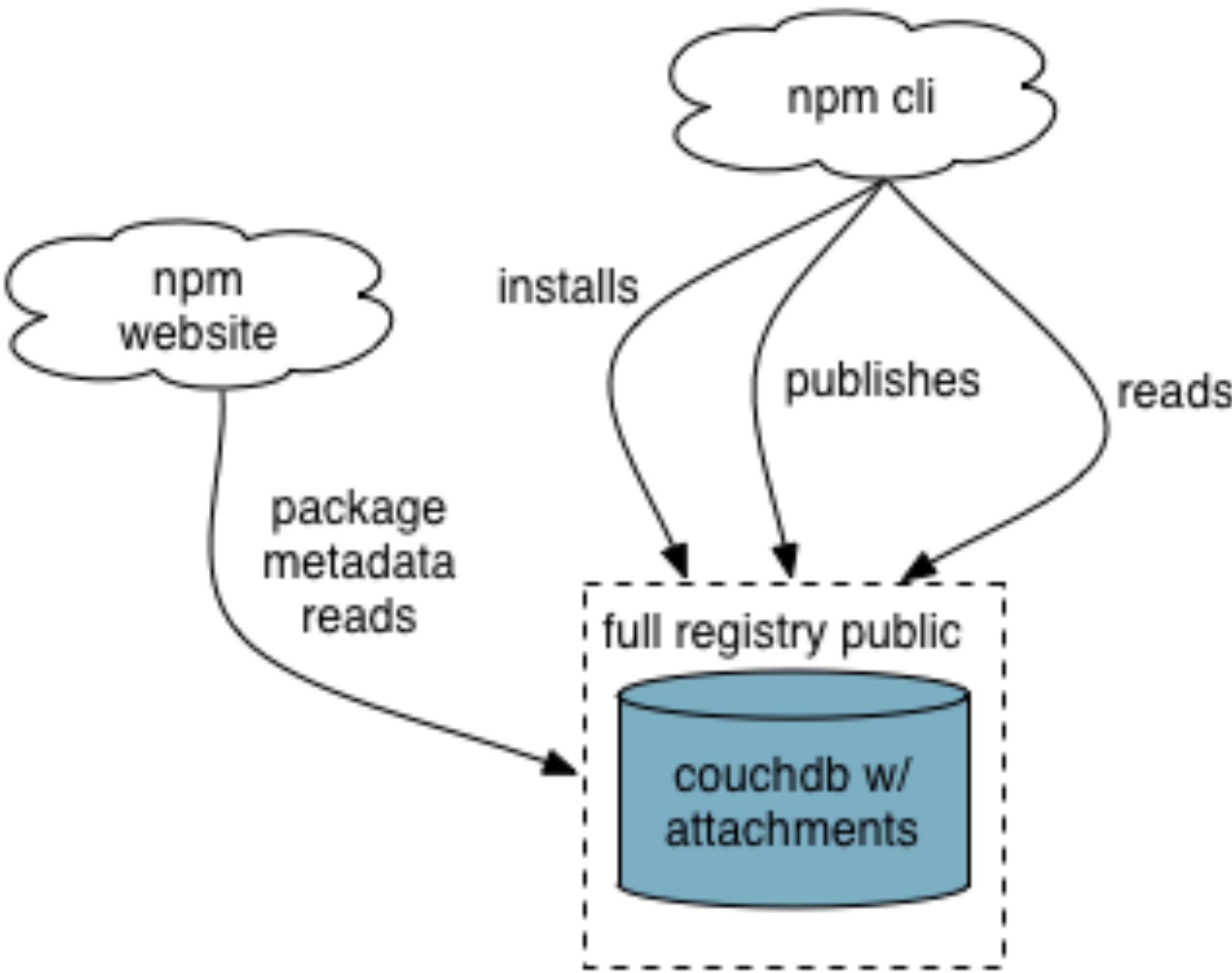
**Q: HOW DO YOU CHEAT?  
A: BY NOT REWRITING THE WHOLE THING.**

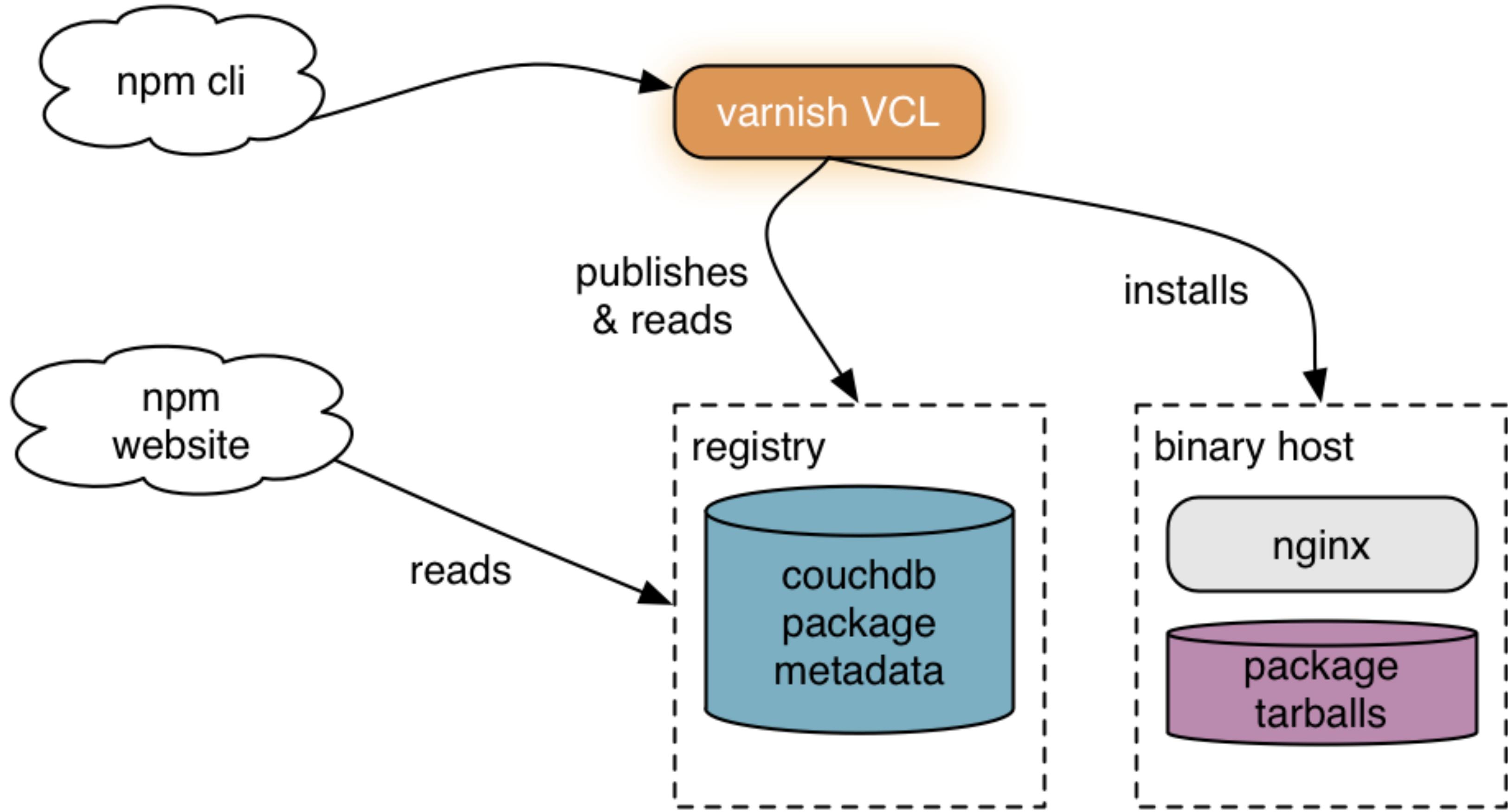
**SLICE OFF A PART OF THE SYSTEM  
INTO A MODULE  
WITH A CLEARLY-DEFINED INTERFACE**

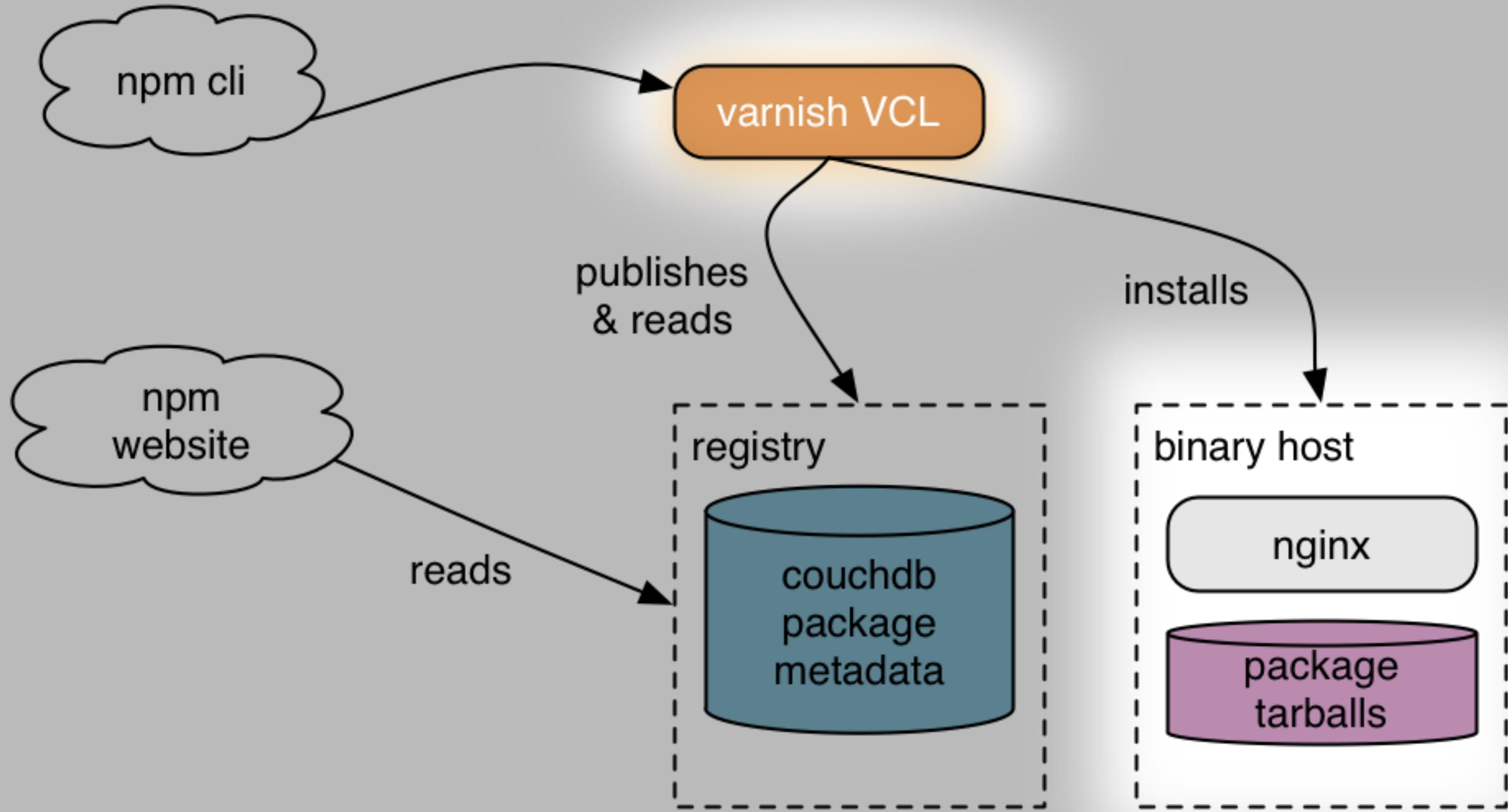
**THEN WRITE A  
SECOND IMPLEMENTATION  
OF THAT INTERFACE**

**SEND REQUESTS TO THAT  
SECOND IMPLEMENTATION  
WITH A PROXY**

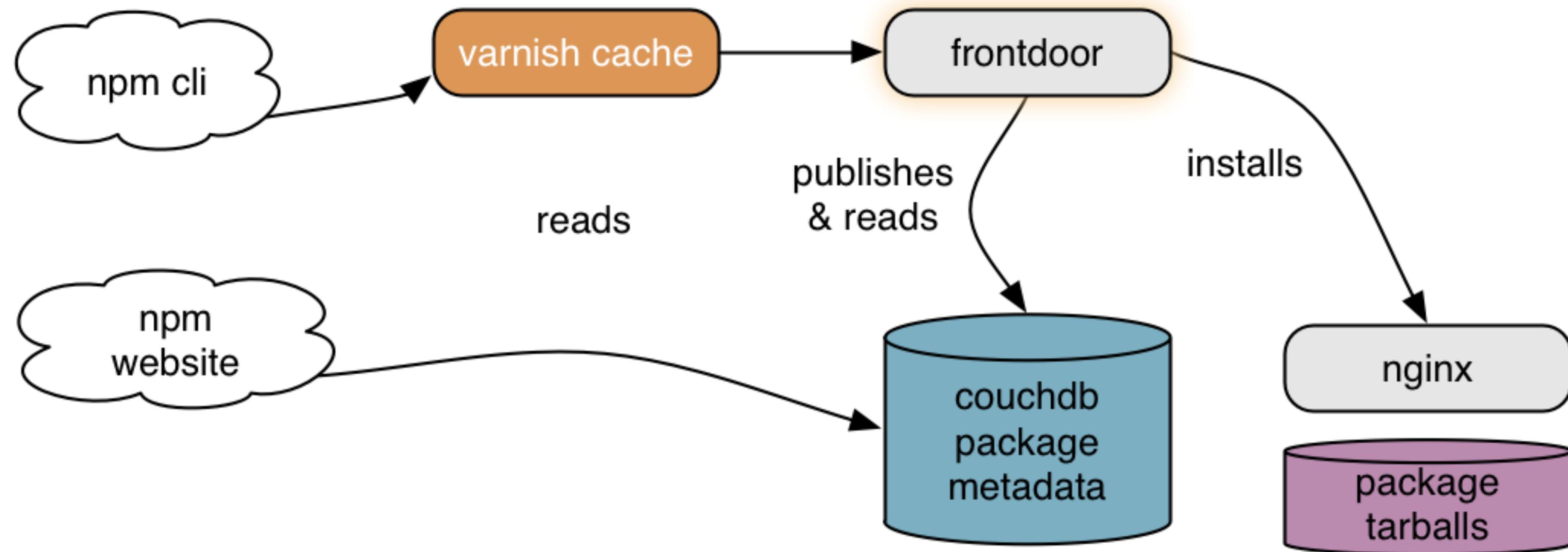


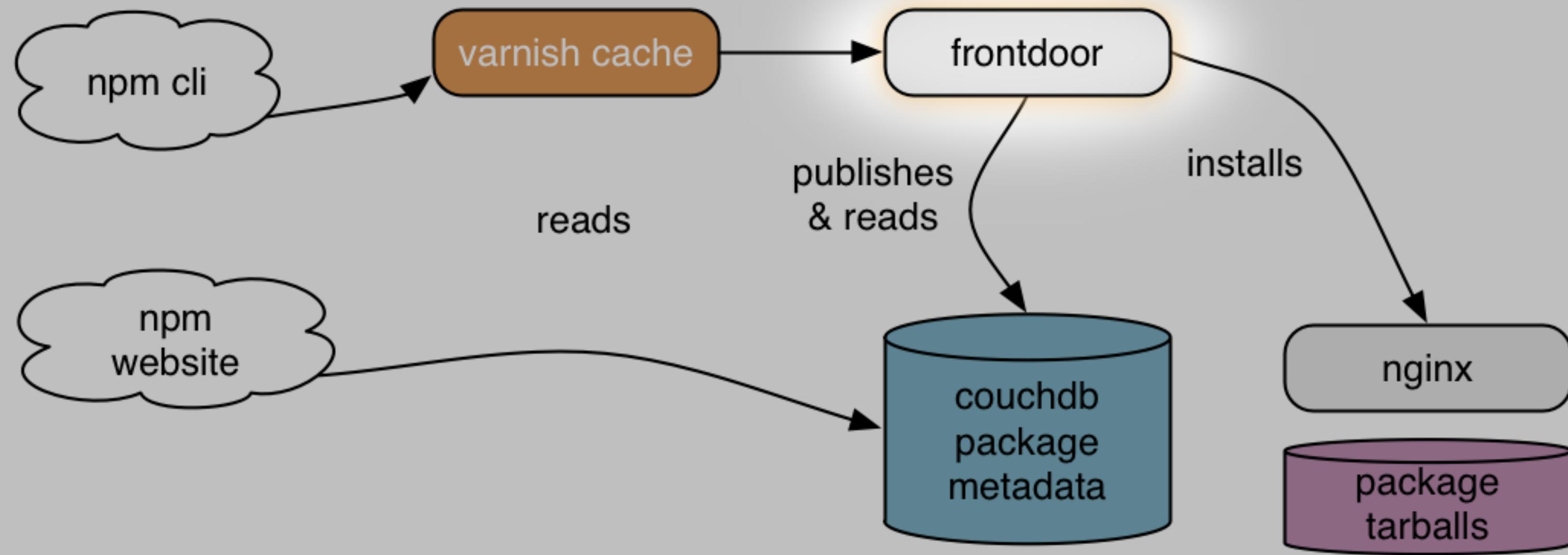






**NOW LET'S MAKE OUR PROXY SMART  
REPLACE VARNISH  
WITH A NODE SERVICE**

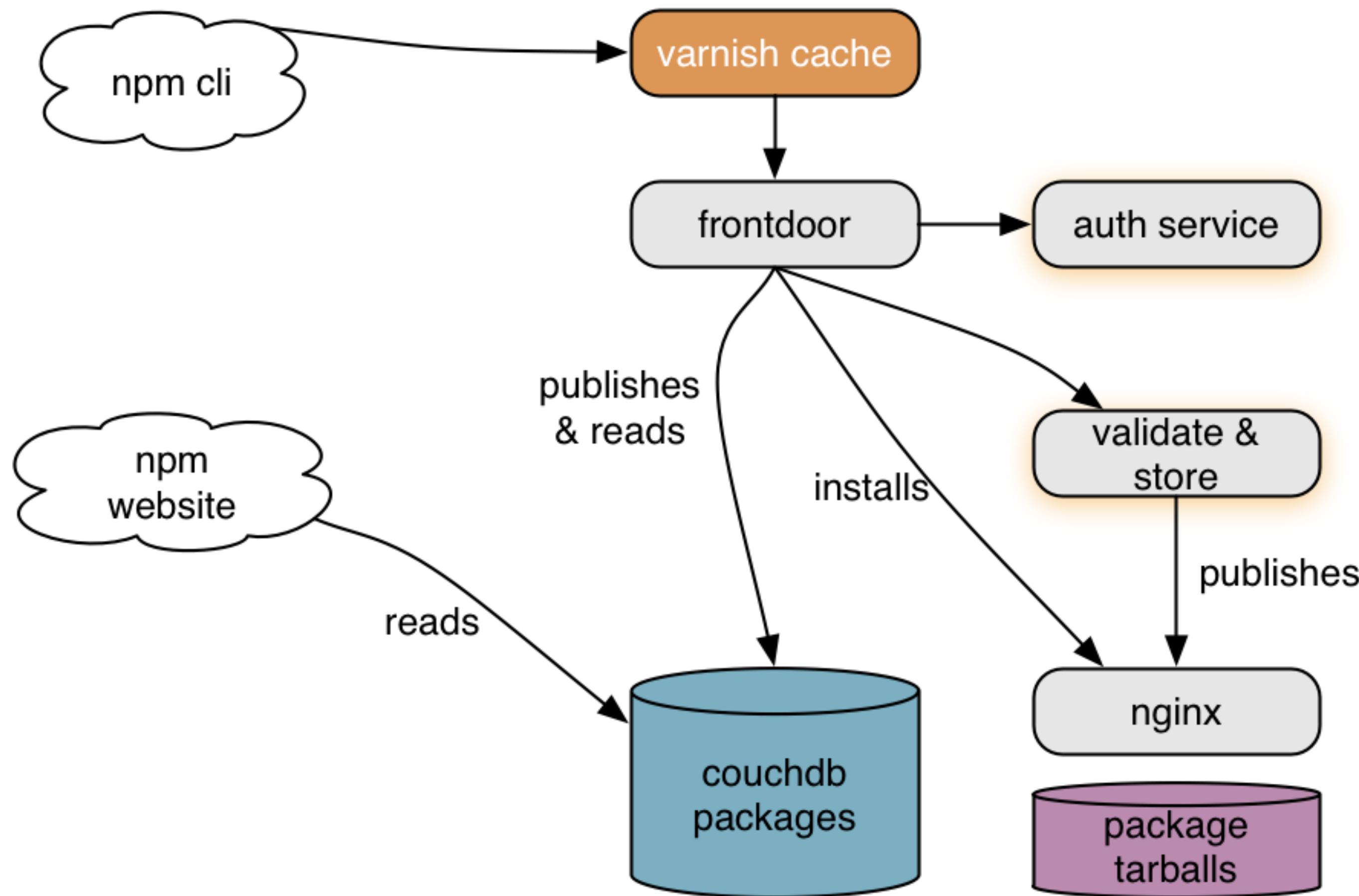


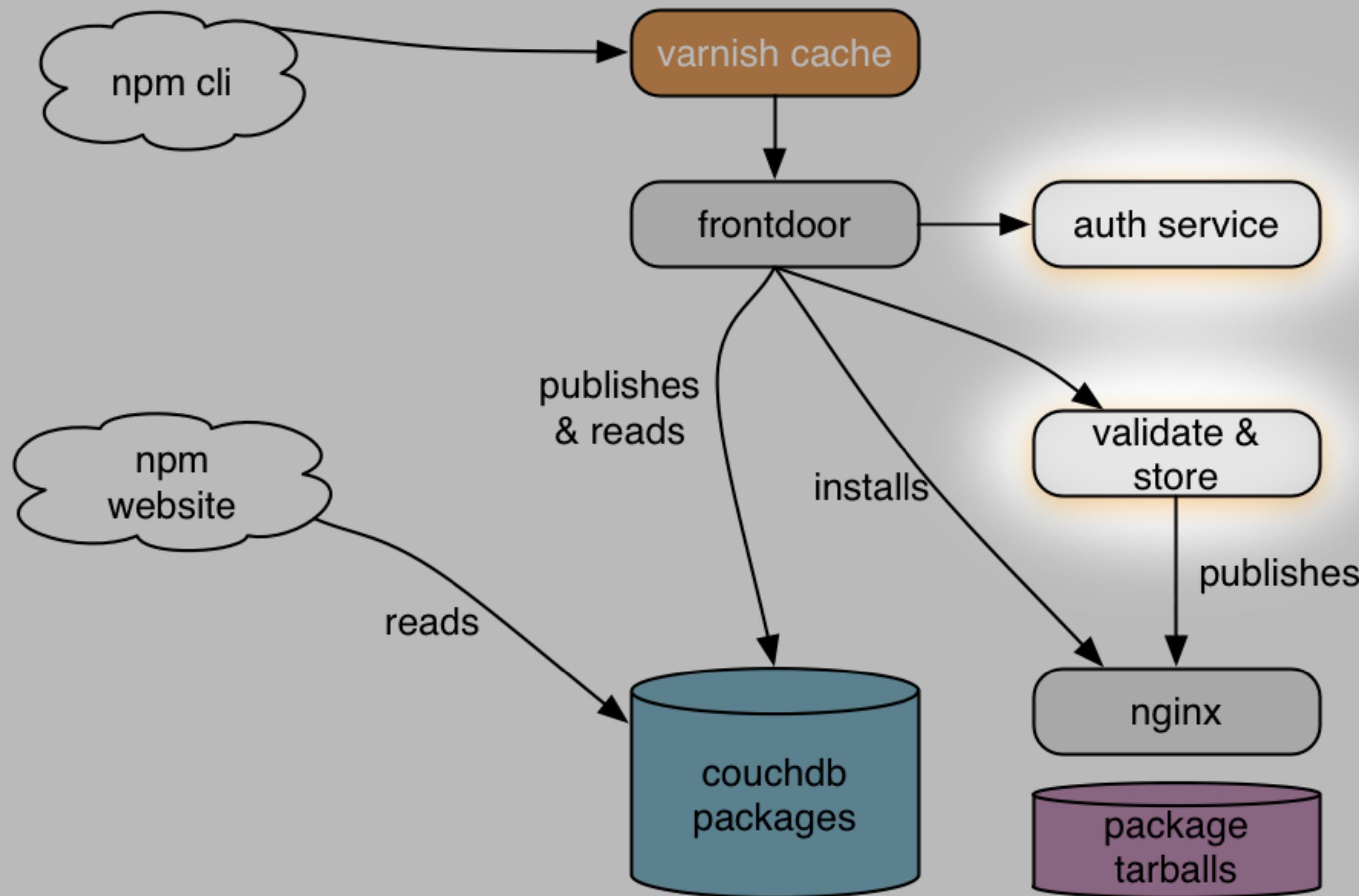


**FIRST STEP  
JUST SEND EVERYTHING THROUGH**

**RECORD WHAT THE PROXY DOES  
HOW DO PEOPLE USE YOUR SERVICE?  
MEASURE PERFORMANCE**

**SECOND STEP  
DIVIDE & CONQUER**



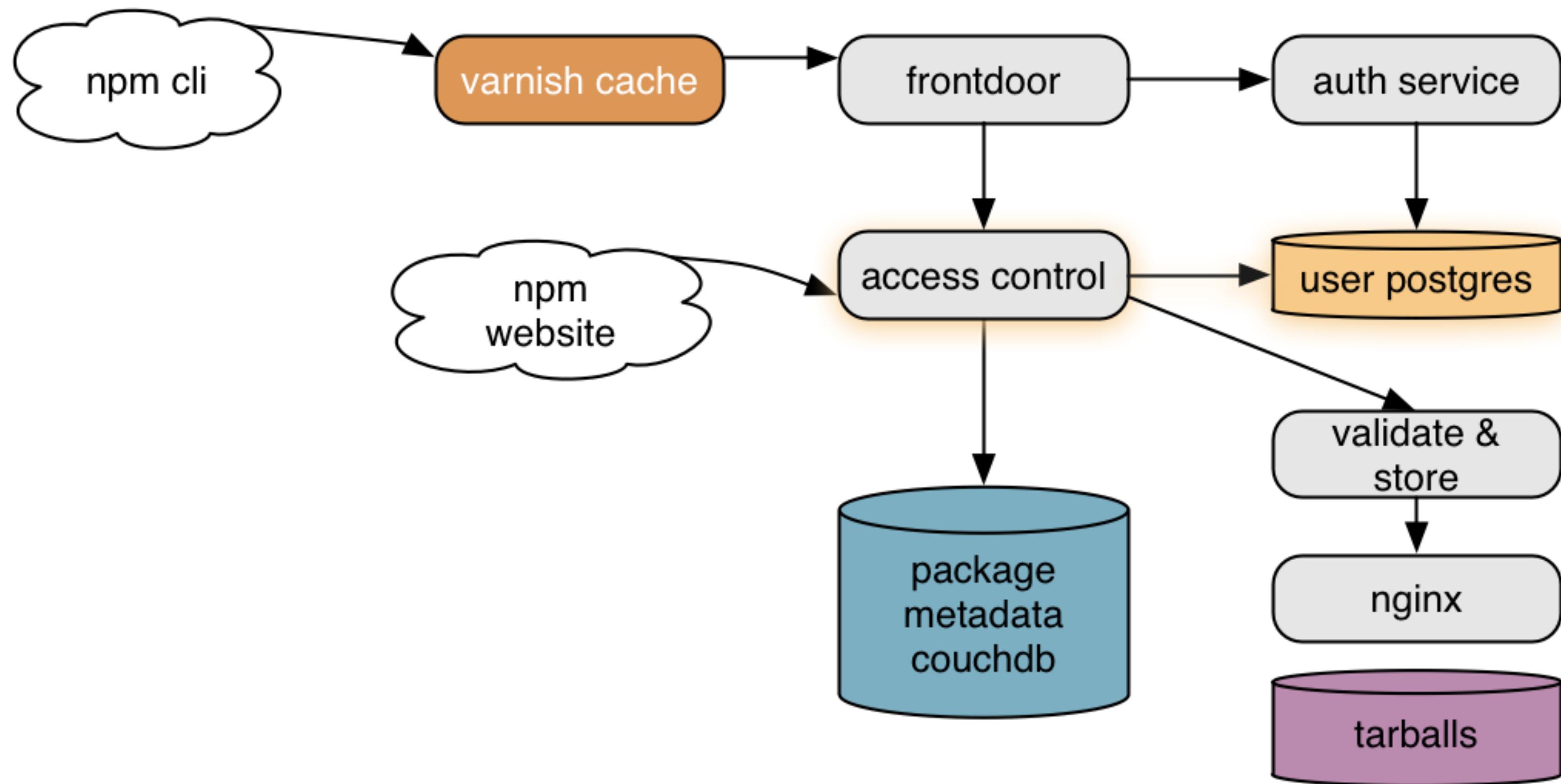


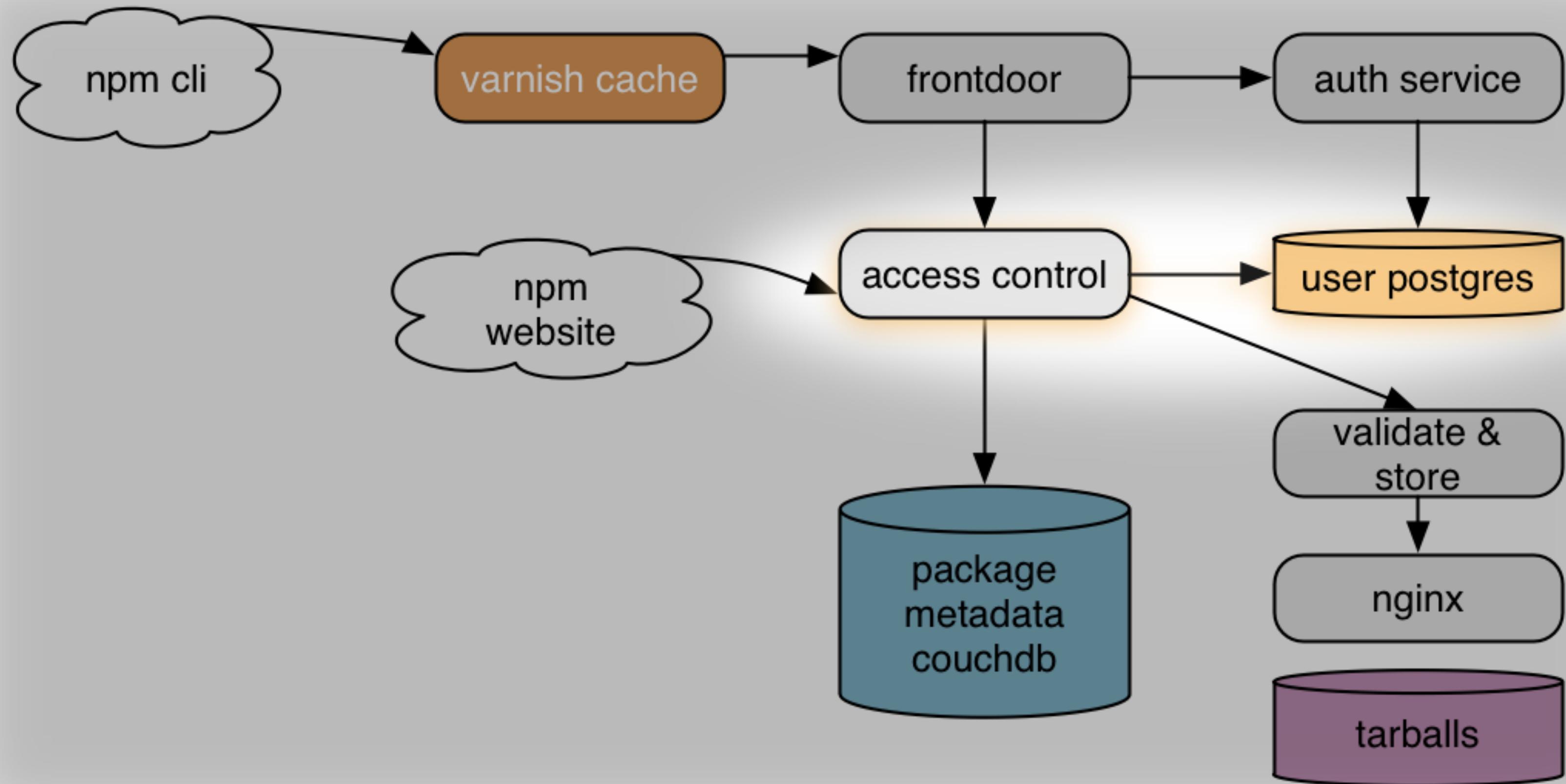
**BEAT GALL'S LAW WITH  
MODULARITY  
AKA INFORMATION HIDING**

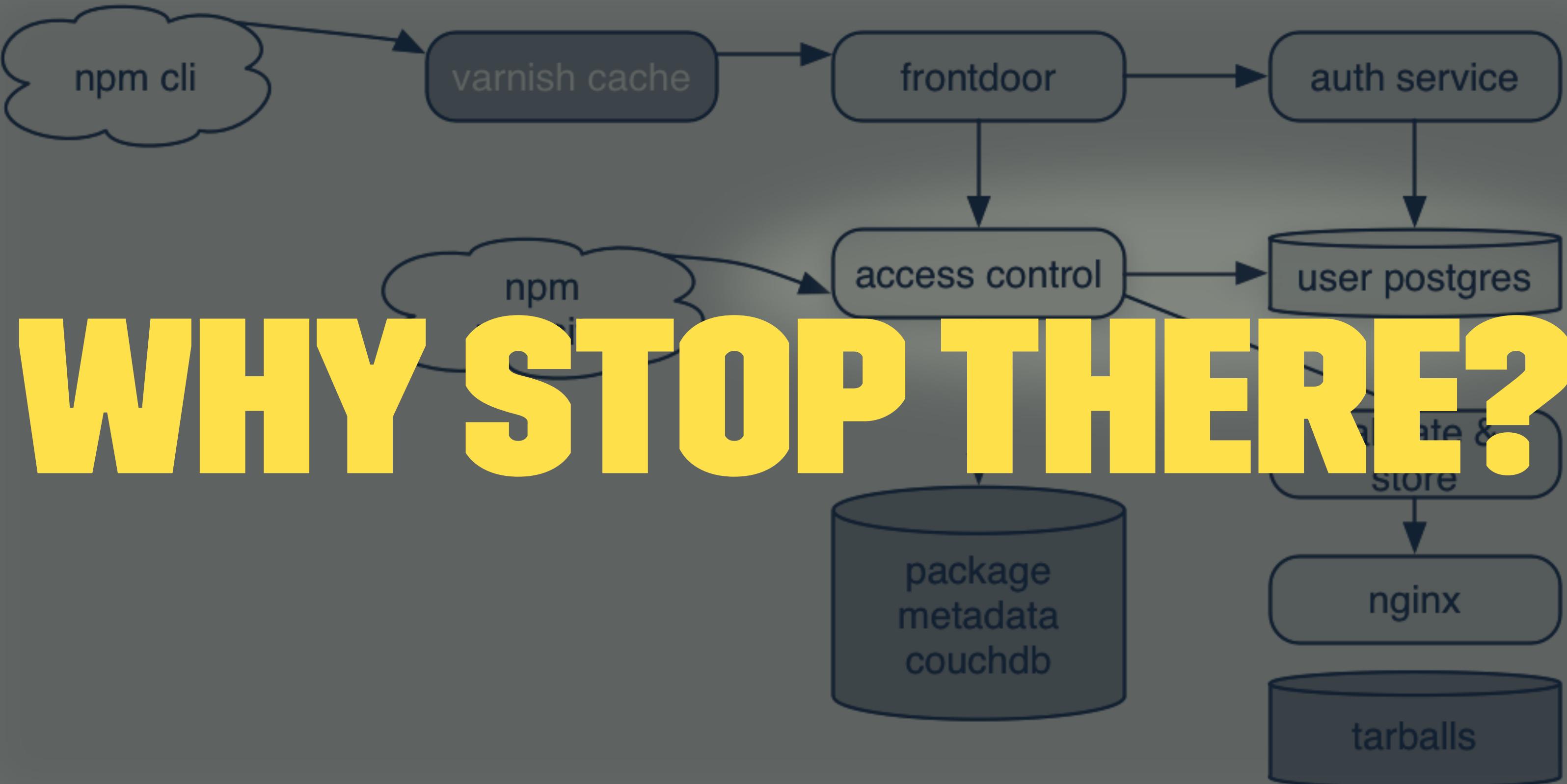
**EACH SERVICE IS A MODULE  
A SIMPLE, TESTABLE SYSTEM  
WHEN VIEWED IN ISOLATION**

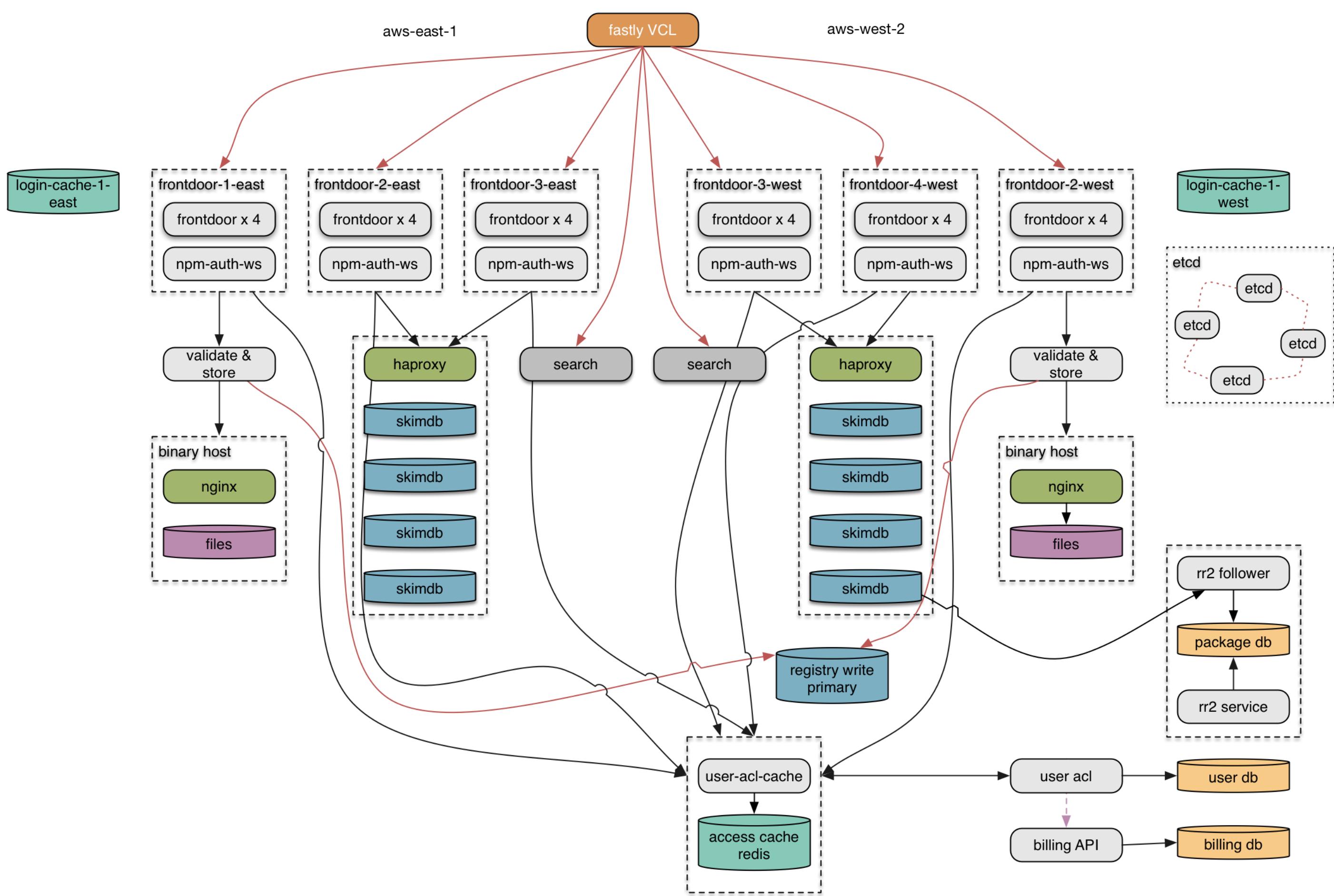
**BEST PART  
NOW YOU CAN CHANGE  
EVERYTHING**

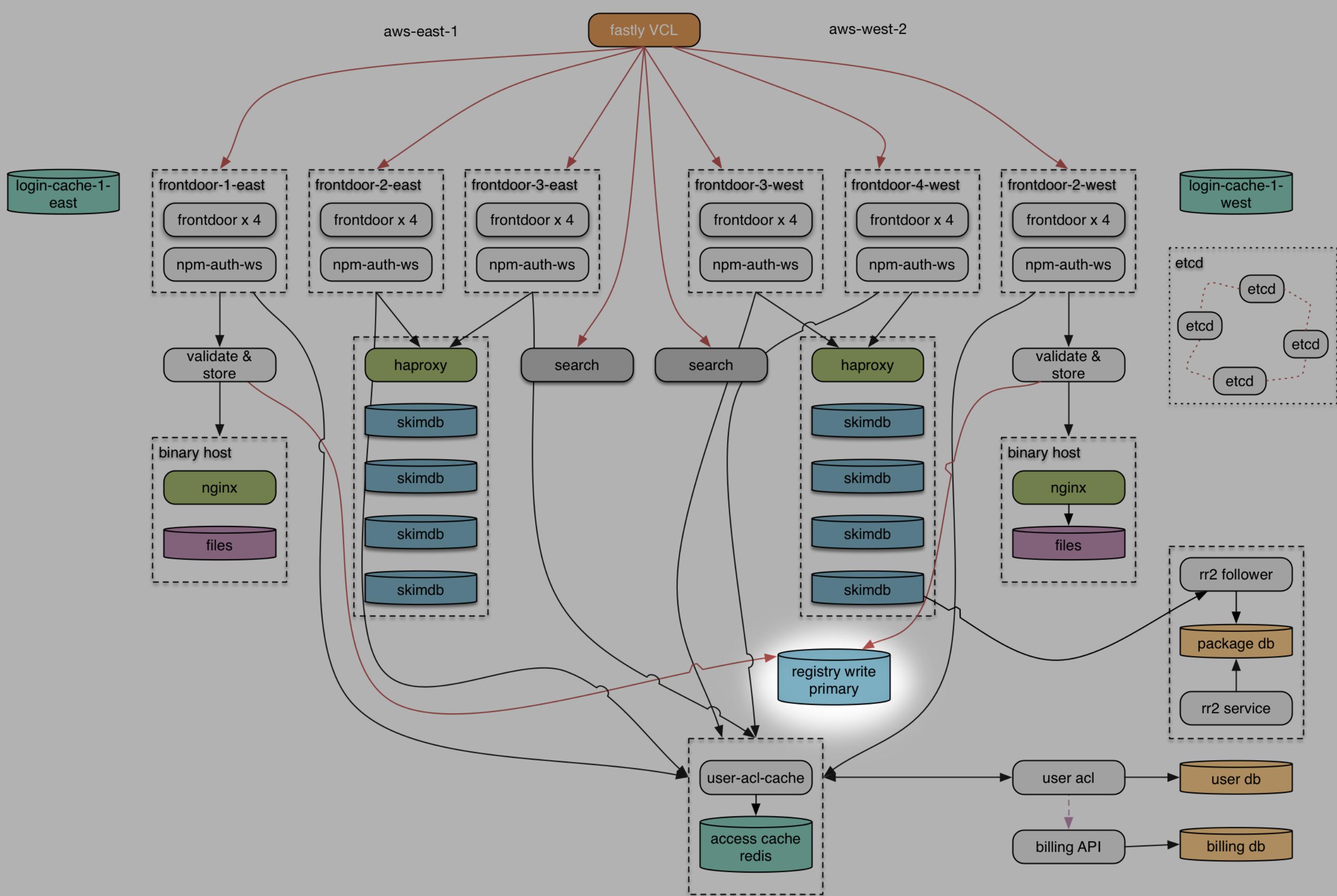
**YOUR PLATFORM  
YOUR DATABASE**











**THE AWESOME**

**EACH PIECE IS A  
SIMPLE SYSTEM**

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

**ISOLATING TASKS  
LET US DEBUG  
EVEN WITHOUT SPLITTING THE MONOLITH**

**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 NOT AWESOME**

**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**

**LET'S RECAP!**  
**HOW DO YOU SURVIVE A REWRITE?**

**SIMPLE WORKING SERVICE FIRST  
SCALE IT LATER**

**RESPECT GALL:  
REWRITE IN PIECES**

**USE A PROXY  
TO DIVIDE & CONQUER**

**BE BOLD**

**YOU CAN CHANGE YOUR SYSTEM**

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

**npm** LOVES YOU  
npm install -g npm@latest